

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky

## DIPLOMOVÁ PRÁCE

2020

Bc. Patrik Koch

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

Virtuální uvedení do provozu za účelem testování a optimalizace  
strojních zařízení  
Virtual Commissioning for Testing and Optimization of Machinery

## Zadání diplomové práce

Student: **Bc. Patrik Koch**  
Studijní program: N2649 Elektrotechnika  
Studijní obor: 2612T041 Řídicí a informační systémy  
Téma: Virtuální uvedení do provozu za účelem testování a optimalizace  
strojních zařízení  
Virtual Commissioning for Testing and Optimization of Machinery  
Jazyk vypracování: čeština

### Zásady pro vypracování:

1. Rozbor možností využití simulace v průmyslové automatizaci.
2. Seznámení a analýza možností virtuálního uvedení do provozu.
3. Návrh a realizace simulačního modelu pro konstrukční model.
4. Návrh a realizace řídicí aplikace.
5. Ověření funkčnosti simulace a řídicí aplikace.
6. Zhodnocení dosažených výsledků.

### Seznam doporučené odborné literatury:

- [1] BERGER, Hans. *Automating with SIMATIC*. 5th edition. Erlangen, Germany: Publicis Publishing, 2013, 284 p. ISBN 978-3895783876.  
[2] Technická dokumentace k použitým systémům a SW Siemens.


Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jiří Koziorek, Ph.D.**

Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020



  
doc. Ing. Jiří Koziorek, Ph.D.  
vedoucí katedry

  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

## Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 30. dubna 2020

  
.....  
podpis studenta

## Poděkování

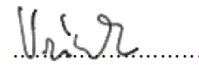
Rád bych poděkoval panu doc. Ing. Jiřímu Koziorkovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této diplomové práce. Taktéž bych rád poděkoval panu Ing. Jakubovi Vránkovi, MBA za trpělivost a umožnění zpracovat diplomovou práci ve firmě ICE Industrial Services. V neposlední řadě bych chtěl také poděkovat své rodině za podporu během celého studia a při tvorbě této závěrečné práce.



## **Prohlášení zástupce spolupracující fyzické osoby**

Souhlasím se zveřejněním této diplomové práce dle požadavku čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TUO Ostrava.

V Praze dne: *30. dubna 2020*



podpis

JAKUB VRÁNEK

## **Abstrakt**

Tato diplomová práce se zabývá virtuálním zprovozněním jednoúčelového stroje pro testování senzorů. Virtuální zprovoznění bylo provedeno za účelem otestování a optimalizace vytvořené řídicí aplikace a případné nalezení mechanických kolizí mezi jednotlivými díly. Výsledkem práce je také zjistit časovou náročnost při vytváření fyzikálně-kinematického a behaviorálního modelu. Virtuální zprovoznění bylo provedeno pomocí simulační techniky Software-in-the-Loop. Řídicí aplikace byla napsána v programovacím jazyce reléových schémat ve firemním standardu MICE a vytvořena, společně s vizualizací, v prostředí TIA Portal V15.1. Pro simulaci řídicího systému byl použit nástroj PLCSIM Advanced 2.0. Behaviorální model byl vytvořen pomocí nástroje SIMIT V9.0 ve verzi Ultimate. Fyzikálně-kinematický model byl realizován pomocí nástroje NX Mechatronics Concept Designer 12.0.1. Všechny simulační modely byly spuštěny na jednom PC a komunikace mezi nimi byla realizována pomocí sdílené paměti. Pro komunikaci mezi automatizačním a behaviorálním modelem byla vytvořena konzolová C# aplikace v IDE Visual Studio. Při virtuálním zprovoznění byl dodržen podobný postup, jako by se jednalo o oživení fyzického stroje. Zprovoznění vedlo k odstranění chyb a optimalizaci řídicí aplikace za pomoci provedených testů na virtuálním stroji.

## **Klíčová slova**

Virtuální uvedení do provozu, Digitální dvojče, Simulace, Model virtuálního stroje, Programovatelný logický automat

## **Abstract**

This master thesis deals with virtual commissioning of one-purpose machinery for sensors testing. Virtual commissioning has been done for testing and optimization of a created control application and possible mechanical collisions detection. The thesis result is also to find out time consumption for the creation of a physical-kinematic and behavioral model. The virtual commissioning has been made using the Software-in-the-Loop simulation technique. The control application and visualization have been made in the TIA Portal V15.1 environment in a ladder programming language using company standard MICE. The simulation tool PLCSIM Advanced 2.0 has been used for the control system simulation. The simulation platform SIMIT V9.0 Ultimate has been used for the electrical and behavioral model design. The physical and kinematic model has been made using tool NX Mechatronics Concept Designer 12.0.1. All of the simulation models have been run on the same PC and used shared memory for data exchange. Communication between the automation and the behavioral model has been made with the help of the C# console application developed in the IDE Visual Studio. The virtual commissioning has been made a similar way as the commissioning of the real machine. The commissioning led to the code error elimination and control application optimization through tests performed on the virtual machine.

## **Key Words**

Virtual Commissioning, Digital Twin, Simulation, Virtual machine model, Programmable logic controller

# Obsah

Seznam použitých symbolů.....	8
Seznam použitých zkratk.....	10
Seznam ilustrací .....	12
Seznam tabulek .....	16
Úvod.....	17
1. Základní pojmy .....	19
1.1 Simulace.....	19
1.2 Kosimulace.....	19
1.3 Simulační techniky .....	20
1.4 Virtuální zprovoznění.....	21
1.5 Digitální dvojče.....	23
1.6 PDM.....	24
1.7 PLM.....	25
2. Simulace v automatizaci.....	26
2.1 Možnosti.....	26
2.2 Úrovně simulace.....	28
2.3 Model virtuálního stroje .....	29
3. Simulační platformy .....	31
3.1 PLC simulátory .....	31
3.2 HMI simulátory .....	32
3.3 Robotické simulátory .....	32
3.4 SIMIT .....	33
3.5 Simcenter Amesim .....	40
3.6 MATLAB.....	41
3.7 FluidSIM .....	41
3.8 NX Mechatronics Concept Designer.....	41
3.9 Tecnomatix Process Simulate .....	47
3.10 Tecnomatix Plant Simulation .....	48
3.11 WinMOD.....	49
3.12 Visual Components .....	51
3.13 ISG-virtuos.....	52

3.14	Delmia V5 .....	52
3.15	CIROS Studio/Education .....	53
3.16	Factory I/O .....	54
3.17	Simulace technologických procesů .....	54
4.	Popis pracoviště.....	56
4.1	Layout .....	57
4.2	Hardware .....	57
4.3	Komunikace .....	59
4.4	Ovládání .....	60
5.	Postup řešení .....	63
5.1	Vytvoření modelu virtuálního stroje .....	63
5.2	Výměna dat .....	64
5.3	Plánované kroky.....	66
6.	Realizace fyzikálně-kinematického modelu.....	69
6.1	Fyzikální vlastnosti .....	70
6.2	Pohyb a senzory .....	72
6.3	Materiál .....	75
6.4	Ostatní zařízení.....	81
6.5	Signály.....	82
6.6	Time-based a Event-based simulace .....	84
7.	Realizace elektrického a behaviorálního modelu .....	85
7.1	Diagramy napájení a prvků rozvaděče .....	87
7.2	Diagramy senzorů a ventilů.....	87
7.3	Diagramy pohonů.....	88
7.4	Diagram měřicího PC.....	96
7.5	Diagramy interakce .....	97
7.6	Diagramy vizualizačních prvků .....	97
7.7	Hlavní přehledový diagram.....	98
8.	Realizace řídicí aplikace.....	99
8.1	HW konfigurace .....	99
8.2	Struktura.....	100
8.3	Symbolika .....	101
8.4	Zpracování vstupů.....	102

8.5	Režimy .....	103
8.6	Akční členy .....	104
8.7	Sekvence .....	106
8.8	Obscné funkce.....	111
8.9	Bezpečnostní program.....	111
8.10	HMI.....	113
9.	Vyhodnocení .....	114
9.1	Nastavení komunikace v SIMIT .....	114
9.2	Nastavení komunikace v NX MCD.....	115
9.3	Vytvoření konzolové aplikace.....	116
9.4	Povolání simulace v TIA Portal .....	119
9.5	Virtuální zprovoznění.....	119
	Závěr .....	124
	Literatura .....	126
	Seznam příloh.....	131

## Seznam použitých symbolů

Symbol	Jednotky	Význam
$G_I$	$A \cdot V^{-1}$	Přenos soustavy s proudem na výstupu
$G_{IN1}$	$rad \cdot s^{-1} \cdot V^{-1}$	Přenos soustavy vnitřní smyčky
$G_{IN2}$	$rad \cdot V^{-1}$	Přenos soustavy prostřední smyčky
$G_{R1}$	-	Přenos regulátoru polohy
$G_{R2}$	-	Přenos regulátoru rychlosti
$G_{R3}$	-	Přenos regulátoru proudu
$G_\theta$	$rad \cdot s^{-1} \cdot V^{-1}$	Přenos soustavy s úhlovou rychlostí na výstupu
$G_\omega$	$rad \cdot V^{-1}$	Přenos soustavy s úhlovou dráhou na výstupu
$i$	A	Proud vinutím
$i_{MG}$	-	Převodový poměr
$I_N$	A	Jmenovitý proud
$J$	$kg \cdot m^2$	Moment setrvačnosti
$J_G$	$kg \cdot m^2$	Moment setrvačnosti převodovky
$J_L$	$kg \cdot m^2$	Moment setrvačnosti zátěže
$J_L$	$kg \cdot m^2 \cdot kg^{-1}$	Moment setrvačnosti na kg užitečné zátěže
$J_M$	$kg \cdot m^2$	Moment setrvačnosti motoru
$J_O$	$kg \cdot m^2$	Moment setrvačnosti standardních sání
$J_S$	$kg \cdot m^2 \cdot m^{-1}$	Moment setrvačnosti na metr zdvihu
$J_W$	$kg \cdot m^2$	Moment setrvačnosti přídatných sání
$k$	-	Počet přídatných sání
$K_e$	$V \cdot rad \cdot s^{-1}$	Konstanta elektromotorického napětí
$K_t$	$Nm \cdot A$	Konstanta momentu motoru
$L$	H	Indukčnost
$l$	m	Pracovní zdvih
$M$	Nm	Moment motoru
$M_H$	Nm	Záběrný moment motoru
$M_{MAX}$	Nm	Maximální moment motoru
$M_N$	Nm	Jmenovitý moment motoru
$n_{MAX}$	$ot \cdot min^{-1}$	Maximální otáčky motoru
$n_N$	$ot \cdot min^{-1}$	Jmenovité otáčky motoru
$P_N$	W	Jmenovitý výkon motoru
$R$	$\Omega$	Odpor
$t_A$	s	Doba rozjetí motoru
$t_B$	s	Doba zastavení motoru
$U$	V	Napájecí napětí
$U_i$	V	Indukované napětí
$U_L$	V	Napětí na cívce
$U_R$	V	Napětí na odporu
$\theta$	rad	Úhlová dráha
$\theta_{MAX}$	rad	Maximální úhlová dráha

$\Phi$	Wb	Magnetický tok
$\omega$	rad·s <sup>-1</sup>	Úhlová rychlost
$\omega_{\text{MAX}}$	rad·s <sup>-1</sup>	Maximální úhlová rychlost



## Seznam použitých zkratk

Zkratka	Význam
<b>2D</b>	Two-Dimensional
<b>3D</b>	Three-Dimensional
<b>ABS</b>	Anti-lock Brake System
<b>API</b>	Application Programming Interface
<b>BOL</b>	Beginning of Life
<b>CAD</b>	Computer-Aided Design
<b>CAE</b>	Computer-Aided Engineering
<b>CAM</b>	Computer-Aided Manufacturing
<b>CIROS</b>	Computer Integrated Robot Simulation
<b>CNC</b>	Computer Numeric Control
<b>CPS</b>	Cyber-Physical System
<b>CPU</b>	Central Processing Unit
<b>DB</b>	Data Block
<b>DCS</b>	Distributed Control System
<b>DI</b>	Digital Input
<b>DO</b>	Digital Output
<b>ECAD</b>	Electronic Computer-Aided Design
<b>EOL</b>	End of Life
<b>F-I/O</b>	Failsafe - Input / Output
<b>FB</b>	Function Block
<b>FC</b>	Function
<b>FHPP</b>	Festo Handling and Positioning Profile
<b>FMI</b>	Functional Mock-up Interface
<b>HiL</b>	Hardware-in-the-Loop
<b>HMI</b>	Human Machine Interface
<b>HW</b>	Hardware
<b>I/O</b>	Input / Output
<b>IDE</b>	Integrated Development Environment
<b>IP</b>	Internet Protocol
<b>IPC</b>	Industrial Personal Computer
<b>IRL</b>	Industrial Robot Language
<b>IT</b>	Information Technology
<b>MCD</b>	Mechatronics Concept Designer
<b>MiL</b>	Model-in-the-Loop
<b>MOL</b>	Middle of Life
<b>NC</b>	Numerical Control
<b>NOK</b>	Not Okay
<b>OB</b>	Organization Block
<b>OK</b>	Okay
<b>OLP</b>	Off-Line Programming

---

<b>OPC DA</b>	OPC Data Access
<b>OPC UA</b>	OPC Unified Architecture
<b>OS</b>	Operating System
<b>PB</b>	Profibus
<b>PC</b>	Personal Computer
<b>PDM</b>	Product Data Management
<b>PID</b>	Proportional Integral Derivative
<b>PiL</b>	Processor-in-the-Loop
<b>PLC</b>	Programmable Logic Controller
<b>PLM</b>	Product Lifecycle Management
<b>PN</b>	Profinet
<b>RCI</b>	Robot Controller Interface
<b>RFID</b>	Radio Frequency Identification
<b>SHM</b>	Shared Memory
<b>SiL</b>	Software-in-the-Loop
<b>SR</b>	Set Reset
<b>SW</b>	Software
<b>VC</b>	Virtual Controller
<b>VR</b>	Virtual Reality
<b>XML</b>	Extensible Markup Language

---

## Seznam ilustrací

Obrázek 1.1:	<i>Kosimulace</i> .....	19
Obrázek 1.2:	<i>V model</i> [7] .....	20
Obrázek 1.3:	<i>Virtuální a klasické zprovoznění</i> [15] .....	21
Obrázek 1.4:	<i>Porovnání workflow při použití virtuálního zprovoznění</i> [16] .....	22
Obrázek 1.5:	<i>Desetinásobný nárůst nákladů s každou fází projektu</i> [17] .....	23
Obrázek 1.6:	<i>Digitální dvojče automatizačního systému a továrny</i> [15] .....	23
Obrázek 1.7:	<i>PDM a PLM</i> [22] .....	24
Obrázek 1.8:	<i>Pět fází životního cyklu výrobku</i> [24] .....	25
Obrázek 1.9:	<i>BOL, MOL a EOL</i> [24] .....	25
Obrázek 2.1:	<i>Simulace výrobní továrny v automotive průmyslu</i> [25] .....	26
Obrázek 2.2:	<i>Simulace ohýbání plechu v prostředí Simufact Forming</i> [28] .....	27
Obrázek 2.3:	<i>Simulace pracoviště s člověkem v prostředí Tecnomatix Process Simulate</i> [29] .....	27
Obrázek 2.4:	<i>Integrace VR do procesu virtuálního zprovoznění</i> [30] .....	28
Obrázek 2.5:	<i>Rozdělení úrovně simulace podle komplexnosti systému</i> [27; 18] .....	28
Obrázek 2.6:	<i>Model virtuálního stroje</i> [18; 27; 31] .....	29
Obrázek 3.1:	<i>Porovnání skutečné a simulované továrny</i> [34] .....	34
Obrázek 3.2:	<i>Příklad diagramu s propojenými komponenty a řídicími prvky</i> [35] .....	35
Obrázek 3.3:	<i>Popis uživatelského prostředí SIMIT</i> [35] .....	37
Obrázek 3.4:	<i>SIMIT – asynchronní pracovní mód</i> [33] .....	37
Obrázek 3.5:	<i>SIMIT – synchronní pracovní mód</i> [33] .....	38
Obrázek 3.6:	<i>SIMIT – pracovní mód synchronní sběrnice</i> [33] .....	38
Obrázek 3.7:	<i>Propojení SIMIT jednotky k SIMIT platformě</i> [33] .....	39
Obrázek 3.8:	<i>Vývojové prostředí Simcenter Amesim – simulace rypadla</i> [39] .....	40
Obrázek 3.9:	<i>Vývojové prostředí FluidSIM – simulace 4 dvojčinných pneumatických válců</i> [41] .....	41
Obrázek 3.10:	<i>Popis uživatelského prostředí NX</i> [44] .....	42
Obrázek 3.11:	<i>Základní nastavení simulace a Physics Engine v prostředí NX</i> .....	46
Obrázek 3.12:	<i>NX Mechatronics Concept Designer workflow</i> [44] .....	46
Obrázek 3.13:	<i>Vývojové prostředí Tecnomatix Process Simulate</i> [46] .....	47
Obrázek 3.14:	<i>Vývojové prostředí Tecnomatix Plant Simulation</i> [49] .....	48
Obrázek 3.15:	<i>Vývojové prostředí WinMOD</i> [52] .....	49
Obrázek 3.16:	<i>Koncept výrobní automatizace ve WinMOD</i> [15] .....	49
Obrázek 3.17:	<i>Koncept procesní automatizace ve WinMOD</i> [15] .....	50
Obrázek 3.18:	<i>Vývojové prostředí Visual Components</i> [55] .....	51
Obrázek 3.19:	<i>Vývojové prostředí ISG Virtuos</i> [59] .....	52
Obrázek 3.20:	<i>Vývojové prostředí DELMIA V5</i> [61] .....	53
Obrázek 3.21:	<i>Vývojové prostředí CIROS Studio</i> [63] .....	53
Obrázek 3.22:	<i>Vývojové prostředí Factory I/O</i> [65] .....	54
Obrázek 4.1:	<i>Model jednoúčelového stroje pro testování senzorů</i> .....	56
Obrázek 4.2:	<i>Layout a rozdělení stanic jednoúčelového stroje pro testování senzorů</i> .....	57
Obrázek 4.3:	<i>Topologie Profinet sítě</i> .....	60
Obrázek 4.4:	<i>Topologie IO-Link sítě</i> .....	60
Obrázek 5.1:	<i>Zvolený model virtuálního stroje pro SiL simulaci</i> .....	63

Obrázek 5.2:	<i>Zvolený model virtuálního stroje pro HiL, případně PiL simulaci .....</i>	64
Obrázek 5.3:	<i>Výměna dat mezi modely pro SiL simulaci se SIMIT V9.0.....</i>	64
Obrázek 5.4:	<i>Výměna dat mezi modely pro HiL nebo PiL simulaci se SIMIT V9.0 .....</i>	64
Obrázek 5.5:	<i>Výměna dat mezi modely pro SiL simulaci se SIMIT V10.1 .....</i>	65
Obrázek 5.6:	<i>Výměna dat mezi modely pro HiL nebo PiL simulaci se SIMIT V10.1 .....</i>	65
Obrázek 5.7:	<i>Zvolený způsob výměny dat .....</i>	66
Obrázek 5.8:	<i>Kroky pro vytvoření modelu v NX MCD .....</i>	66
Obrázek 5.9:	<i>Kroky pro vytvoření modelu v SIMIT.....</i>	67
Obrázek 5.10:	<i>Kroky pro vytvoření řídicí aplikace v TIA Portal.....</i>	68
Obrázek 5.11:	<i>Kroky pro virtuální zprovoznění.....</i>	68
Obrázek 6.1:	<i>Importovaný CAD model stroje do prostředí NX.....</i>	69
Obrázek 6.2:	<i>Importovaný CAD model senzoru do prostředí NX.....</i>	69
Obrázek 6.3:	<i>Odhalený model stroje v prostředí NX.....</i>	70
Obrázek 6.4:	<i>Přidání nedeformovatelného tělesa v prostředí NX .....</i>	70
Obrázek 6.5:	<i>Přidání posuvného spoje v prostředí NX.....</i>	71
Obrázek 6.6:	<i>Přidání kolizního tělesa v prostředí NX .....</i>	71
Obrázek 6.7:	<i>Přidání prvku řízení polohy v prostředí NX .....</i>	72
Obrázek 6.8:	<i>Přidání koncového spínače polohy v prostředí NX.....</i>	72
Obrázek 6.9:	<i>Přidání prvku virtuální osa a řízení rychlosti v prostředí NX.....</i>	73
Obrázek 6.10:	<i>Přidání prvku elektronická vačka v prostředí NX .....</i>	73
Obrázek 6.11:	<i>Převodní charakteristika mezi rotorem a hřídelí v prostředí NX.....</i>	74
Obrázek 6.12:	<i>Převodní charakteristika mezi hřídelí a karuselem v prostředí NX .....</i>	74
Obrázek 6.13:	<i>Přidání kolizního senzoru v prostředí NX .....</i>	75
Obrázek 6.14:	<i>Lůžko se založeným materiálem v prostředí NX.....</i>	75
Obrázek 6.15:	<i>Přidání zdroje objektů v prostředí NX .....</i>	76
Obrázek 6.16:	<i>Přidání zániku objektu v prostředí NX.....</i>	76
Obrázek 6.17:	<i>Přidání kolizních těles na materiál v prostředí NX.....</i>	77
Obrázek 6.18:	<i>Přidání kolizních těles na lůžka v prostředí NX.....</i>	77
Obrázek 6.19:	<i>Přidání adaptéru signálu pro generování a odstraňování materiálu v prostředí NX.....</i>	78
Obrázek 6.20:	<i>Přidání pevného spoje na manipulátor v prostředí NX.....</i>	78
Obrázek 6.21:	<i>Přidání kolizních senzorů na přísavky v prostředí NX.....</i>	79
Obrázek 6.22:	<i>Vytvoření signálu pro aktivaci vakua v prostředí NX .....</i>	79
Obrázek 6.23:	<i>Vytvoření operací pro uchycení/uvolnění materiálu k manipulátoru v prostředí NX.....</i>	80
Obrázek 6.24:	<i>Nastavení kolizního tělesa a kolizního senzoru – čtečka 2D kódů .....</i>	80
Obrázek 6.25:	<i>Nastavení kolizního tělesa a kolizního senzoru – pyrometr .....</i>	81
Obrázek 6.26:	<i>Přidání měniče zobrazení na maják a tlačítko v prostředí NX.....</i>	81
Obrázek 6.27:	<i>Simulace pohybu energetického řetězu v prostředí NX.....</i>	82
Obrázek 6.28:	<i>Simulace pohybu dveří v prostředí NX.....</i>	82
Obrázek 6.29:	<i>Přidání signálu na osu X manipulátoru v prostředí NX.....</i>	83
Obrázek 6.30:	<i>Seznam vytvořených signálů v prostředí NX .....</i>	83
Obrázek 6.31:	<i>Export signálů z prostředí NX.....</i>	84
Obrázek 6.32:	<i>Ganttův diagram pro vytváření sekvencí v prostředí NX.....</i>	84
Obrázek 7.1:	<i>Modifikace hlavičky exportovaných signálů z NX MCD.....</i>	85
Obrázek 7.2:	<i>Import signálů z NX do SIMIT .....</i>	85

Obrázek 7.3:	<i>Importované signály z NX v SIMIT</i> .....	86
Obrázek 7.4:	<i>Nastavení simulace v SIMIT</i> .....	86
Obrázek 7.5:	<i>Část diagramu simulující prvky rozvaděče v SIMIT</i> .....	87
Obrázek 7.6:	<i>Diagram senzoru v SIMIT</i> .....	87
Obrázek 7.7:	<i>Diagram ventilu v SIMIT</i> .....	88
Obrázek 7.8:	<i>Diagram motoru karuselu v SIMIT</i> .....	89
Obrázek 7.9:	<i>Náhradní schéma DC motoru [69]</i> .....	89
Obrázek 7.10:	<i>Model DC motoru</i> .....	90
Obrázek 7.11:	<i>Diagram lineárního DC motoru manipulátoru v SIMIT</i> .....	92
Obrázek 7.12:	<i>Zpracování I/O signálu pro CMMO jednotku v SIMIT</i> .....	92
Obrázek 7.13:	<i>Část logiky zpracování signálu CMMO jednotky v SIMIT</i> .....	93
Obrázek 7.14:	<i>Blokové schéma kaskádového regulačního obvodu</i> .....	93
Obrázek 7.15:	<i>Diagram kaskádového zapojení regulátorů CMMO jednotky v SIMIT</i> .....	94
Obrázek 7.16:	<i>Část diagramu chování měřícího PC v SIMIT</i> .....	96
Obrázek 7.17:	<i>Diagram operací s materiálem v SIMIT</i> .....	97
Obrázek 7.18:	<i>Diagram majáku v SIMIT</i> .....	97
Obrázek 7.19:	<i>Hlavní přehledový diagram v SIMIT</i> .....	98
Obrázek 8.1:	<i>Hardware konfigurace PLC v TIA Portal</i> .....	99
Obrázek 8.2:	<i>Hardware konfigurace Profinet sítě v TIA Portal</i> .....	100
Obrázek 8.3:	<i>Struktura programu v TIA Portal</i> .....	101
Obrázek 8.4:	<i>Hlavní funkce programu OB1 v TIA Portal</i> .....	101
Obrázek 8.5:	<i>Vytvoření a importované PLC tagy v TIA Portal</i> .....	102
Obrázek 8.6:	<i>Datové bloky stanice 120 v TIA Portal</i> .....	102
Obrázek 8.7:	<i>Funkce zpracování vstupů v TIA Portal</i> .....	103
Obrázek 8.8:	<i>Zpracování vstupů IO-Link slave modulu 120KF11 v TIA Portal</i> .....	103
Obrázek 8.9:	<i>Řízení režimů zóny a stanice v TIA Portal</i> .....	104
Obrázek 8.10:	<i>Logika pro uvolnění a interlock pohybu ventilu v TIA Portal</i> .....	104
Obrázek 8.11:	<i>Logika pro automatický pohyb a zpracování senzoru v TIA Portal</i> .....	105
Obrázek 8.12:	<i>Část FB pro řízení ventilu v TIA Portal</i> .....	105
Obrázek 8.13:	<i>Sekvenční FB v TIA Portal</i> .....	106
Obrázek 8.14:	<i>Podmínky přechodu do následujícího kroku v sekvenčním FB v TIA Portal</i> .....	106
Obrázek 8.15:	<i>Vyhodnocení sekvenčních alarmů v sekvenčním FB v TIA Portal</i> .....	107
Obrázek 8.16:	<i>Sekvence stanice 120 v TIA Portal</i> .....	108
Obrázek 8.17:	<i>Sekvence stanice 130 v TIA Portal</i> .....	110
Obrázek 8.18:	<i>Funkce pro komunikaci s měřícím PC v TIA Portal</i> .....	111
Obrázek 8.19:	<i>Struktura bezpečnostního programu v TIA Portal</i> .....	111
Obrázek 8.20:	<i>Zpracování signálu e-stopky v TIA Portal</i> .....	112
Obrázek 8.21:	<i>Kontrola zpětné vazby při sepnutí stykače karuselu v TIA Portal</i> .....	112
Obrázek 8.22:	<i>Přehledová obrazovka HMI v TIA Portal</i> .....	113
Obrázek 9.1:	<i>Vytvoření propojení SHM v prostředí SIMIT</i> .....	114
Obrázek 9.2:	<i>Nastavení propojení SHM s NX v prostředí SIMIT</i> .....	114
Obrázek 9.3:	<i>Nastavení propojení SHM se C# aplikací v prostředí SIMIT</i> .....	114
Obrázek 9.4:	<i>Externí konfigurace signálů v prostředí NX</i> .....	115
Obrázek 9.5:	<i>Napojení signálů v prostředí NX</i> .....	116

Obrázek 9.6:	<i>Přehled funkčního principu PLCSIM Advanced [74].....</i>	116
Obrázek 9.7:	<i>Struktura paměťové oblasti [35].....</i>	117
Obrázek 9.8:	<i>Globální proměnné aplikace .....</i>	117
Obrázek 9.9:	<i>Hlavní funkce aplikace Main.....</i>	118
Obrázek 9.10:	<i>Spuštěná aplikace s úspěšně vytvořenou komunikací.....</i>	118
Obrázek 9.11:	<i>Otevření vlastností projektu TIA Portal .....</i>	119
Obrázek 9.12:	<i>Povolení simulace v projektu TIA Portal .....</i>	119
Obrázek 9.13:	<i>Vytvoření instance PLCSIM Advanced .....</i>	120
Obrázek 9.14:	<i>Řízení stroje pomocí simulovaného HMI .....</i>	121
Obrázek 9.15:	<i>Monitorování řídicí aplikace během testů.....</i>	121
Obrázek 9.16:	<i>Virtuálně zprovozněný stroj v automatickém režimu.....</i>	122

## Seznam tabulek

Tabulka 3.1:	<i>Zařazení vybraných nástrojů podle modelu virtuálního stroje a úrovně simulace ...</i>	31
Tabulka 3.2:	<i>Seznam simulátoru PLC .....</i>	32
Tabulka 3.3:	<i>Seznam simulátoru HMI .....</i>	32
Tabulka 3.4:	<i>Seznam simulátorů robotů .....</i>	33
Tabulka 3.5:	<i>Typy řídicích prvků [33] .....</i>	35
Tabulka 3.6:	<i>Typy grafických prvků [33] .....</i>	36
Tabulka 3.7:	<i>Typy propojení se SIMIT [33] .....</i>	39
Tabulka 3.8:	<i>Fyzikální vlastnosti v NX MCD [44] .....</i>	43
Tabulka 3.9:	<i>Typy spojů v NX MCD [44] .....</i>	43
Tabulka 3.10:	<i>Akční členy v NX MCD [44] .....</i>	44
Tabulka 3.11:	<i>Senzory v NX MCD [44] .....</i>	45
Tabulka 3.12:	<i>Seznam nástrojů pro simulaci specifických technologických procesů .....</i>	55
Tabulka 4.1:	<i>Výměna signálu mezi PLC a měřícím PC (z pohledu PLC) .....</i>	60
Tabulka 7.1:	<i>Parametry modelu motoru .....</i>	91
Tabulka 7.2:	<i>Seznam výměny signálu s jednotkou CMMO-ST [72] .....</i>	92
Tabulka 7.3:	<i>Parametry regulátorů CMMO jednotky .....</i>	96
Tabulka 7.4:	<i>Ovládání osvětlení majáku .....</i>	98
Tabulka 8.1:	<i>Hardware konfigurace PLC a adresy .....</i>	99
Tabulka 8.2:	<i>Hardware konfigurace HMI .....</i>	100
Tabulka 8.3:	<i>IP adresy a Profinet jména síťových zařízení .....</i>	100
Tabulka 8.4:	<i>Výchozí stav stanice 120 .....</i>	107
Tabulka 8.5:	<i>Pracovní cyklus stanice 120 .....</i>	108
Tabulka 8.6:	<i>Výchozí stav stanice 130 .....</i>	109
Tabulka 8.7:	<i>Pracovní cyklus stanice 130 .....</i>	109
Tabulka 9.1:	<i>Časová náročnost při vytváření simulačních modelů .....</i>	123

# Úvod

Průmysl 4.0 či čtvrtá průmyslová revoluce je v poslední době často zmiňovaný pojem. Jedná se o současný trend digitalizace a automatizace výroby. Společně s jeho představením se začalo používat spousta dalších pojmů, jako jsou např. chytrá továrna, internet věcí, prediktivní údržba, digitální dvojče nebo virtuální uvedení do provozu. Tato závěrečná práce je úzce spojena s posledními dvěma výrazy, s kterými také úzce souvisí simulace systému.

Nezbytným teoretickým podkladem pro zpracování této práce je vysvětlení základních pojmů simulace a virtuálního zprovoznění. Jsou zde zmíněny možnosti simulace systému průmyslové automatizace, rozdělení simulace v průmyslové automatizaci dle rozsahu simulovaného systému a je zde vysvětlen model virtuálního stroje. Následuje kapitola s popisem vybraných SW nástrojů vhodných pro virtuální zprovoznění stroje. Dle získaných informací jsou tyto simulační nástroje zařazeny podle úrovně simulace a podle modelu virtuálního stroje.

Za účelem virtuálního zprovoznění byl firmou ICE Industrial Services poskytnut skutečný model již sestaveného jednoúčelového stroje. Stroj slouží pro testování funkčnosti senzorů pro měření teploty, hladiny a přítomnosti Ad-blue v nádržích u dieselových aut. Informace týkající se popisu stroje a jeho požadavků na ovládání jsou v této práci patřičně uvedeny. Pro oživení daného stroje byl získán jeho konstrukční model včetně modelu testovaného senzoru.

Na začátku praktické části je stanoven postup řešení. Nejdříve je složen model virtuálního stroje pomocí vybraných nástrojů a jsou rozebrány jejich vzájemné komunikační schopnosti. Model virtuálního stroje je vytvořen ve dvou variantách, v závislosti na použití skutečného nebo simulovaného PLC. S ohledem na použité nástroje jsou sestaveny kroky pro vytvoření fyzikálně-kinematického, elektrického a behaviorálního a automatizačního modelu a kroky pro závěrečné virtuální zprovoznění stroje.

Následující kapitola se zabývá realizací fyzikálně-kinematického modelu v prostředí NX Mechatronics Concept Designer V12.0.1. Na statický model stroje a materiálu jsou aplikovány fyzikální vlastnosti. Jsou zde určeny pohyblivé a kolizní části stroje, vytvořeny potřebné vazby pro manipulaci s materiálem a převodní charakteristiky mezi mechanickými částmi stroje. Závěrem je vytvoření time-based simulace vedoucí k ověření funkčnosti fyzikálně-kinematického modelu.

Pro vytvoření elektrického a behaviorálního modelu je použit nástroj SIMIT V9.0. Na základě elektro-dokumentace je navržen diagram chování elektro a pneumatických rozvodů. Pro simulaci chování pohonů, elektropneumatických ventilů a senzorů jsou vytvořena makra, umožňující tvořit instance daného makra. Pro řízení polohy lineárního pohonu je vyhotoven model chování řídicí jednotky a model DC motoru, jehož pozice je regulována za pomoci kaskádní regulace. Parametry regulátoru jsou navrženy pomocí nástroje MATLAB Simulink. Z důvodu vyhodnocení měření na dílu je také vytvořeno chování měřicího PC. Za účelem interakce a sledování důležitých signálů slouží hlavní přehledový diagram.

Pro řízení stroje je vytvořena řídicí aplikace v prostředí TIA Portal V15.1 a pro simulaci PLC je využito nástroje PLCSIM Advanced 2.0. Aplikace je vytvořena ve firemním standardu MICE, na základě kterého, je dodrženo standardizovaného pojmenování proměnných a strukturovaného rozdělení



jednotlivých části programu. Režimy, řízení akčních členů, zpracování vstupů a generování alarmových hlášek je řešeno pomocí standardizovaných funkčních bloků. Sekvenční části programu stanic jsou naprogramovány z připravených tabulek pracovního cyklu dané stanice. Závěrem je vytvořena bezpečnostní část programu a vizualizační aplikace.

Finální kapitola se zabývá řešením nastavení komunikace mezi jednotlivými nástroji. Z důvodu nepodporované výměny dat, pomocí přímého spojení, mezi instancí PLCSIM Advanced a SIMIT V9.0 je naprogramována konzolová C# aplikace zajišťující vzájemnou komunikaci. V poslední části je zprovozněn virtuální stroj a otestována řídicí aplikace podobně jako u skutečného stroje. Na závěr je práce zhodnocena z hlediska časové náročnosti při vytváření simulačních modelů a jsou zmíněny získané poznatky.

# 1. Základní pojmy

Tato kapitola slouží jako vstupní brána do dané oblasti. V jednotlivých podkapitolách jsou vysvětleny základní pojmy týkající se simulace a virtuálního zprovoznění. Na konci kapitoly je také popsáno, co je to životní cyklus výrobku a jeho souvislost s probíranou problematikou.

## 1.1 Simulace

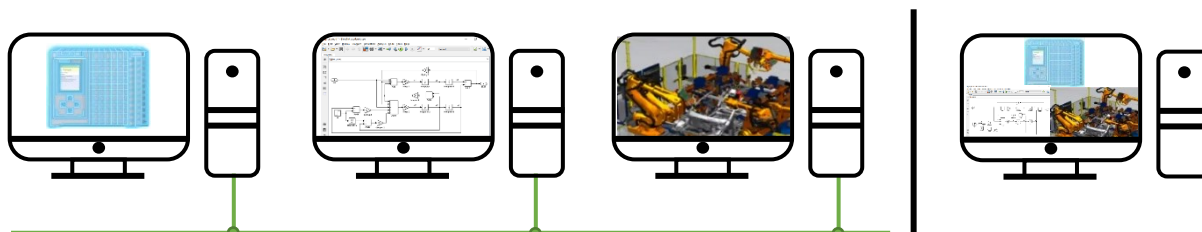
Pojem simulace může být v literatuře různě vysvětlen. Nejčastěji je popsán jako napodobení skutečného systému nebo procesu pomocí modelu systému závislého na čase, prostřednictvím kterého dokážeme nalézt vady a problémy vyskytující se v daném systému a přispět tak k jejich včasnému odstranění. Jakékoliv napodobování systému nebo procesu může být tedy nazýváno jako simulace. Cílem simulace je pomocí vytvořeného simulačního modelu zkoumat chování systému. Tento model má obvykle podobu souborů s předpoklady týkajícími se fungování systému. Tyto předpoklady jsou vyjádřeny v matematických, logických a symbolických vztazích mezi entitami nebo předměty zájmu systému. V případě, že je systém jednoduchý, jeho model může být popsán a vyřešen analyticky. [1; 2; 3]

### 1.1.1 Výhody a nevýhody

Simulace je modelování, díky kterému můžeme analyzovat rozsáhlé složité systémy. Jako největší výhoda simulace je považována úspora financí, odpadá totiž nutnost sledovat a zkoumat reálný systém před jeho uvedením do provozu. Snadno dokážeme analyzovat komplexní systémy. Při simulaci se používají skutečná data, a proto jsou výsledky velmi přiblížené k realitě. Snadnou změnou podmínek v simulaci dokážeme jednoduše zkoumat reakce výstupů systému. V kritických situacích, kde by testování na reálném systému mohlo být dosti nákladné nebo i dokonce nebezpečné, mohou být zkoumány bez jakéhokoliv rizika. Pro podrobné prozkoumání systému můžeme jeho chování sledovat dlouhodobě s možností zrychlení nebo zpomalení simulace. [1]

Mezi nevýhody simulace patří vyšší počáteční náklady potřebné k vytvoření simulačního modelu. Také může být nákladné měřit vzájemný vliv mezi dvěma proměnnými. Během vytváření simulace je nezbytné porozumět skutečnému fyzickému systému a také je zapotřebí ovládat znalosti různých parametrů systému. [1]

## 1.2 Kosimulace



Obrázek 1.1: Kosimulace

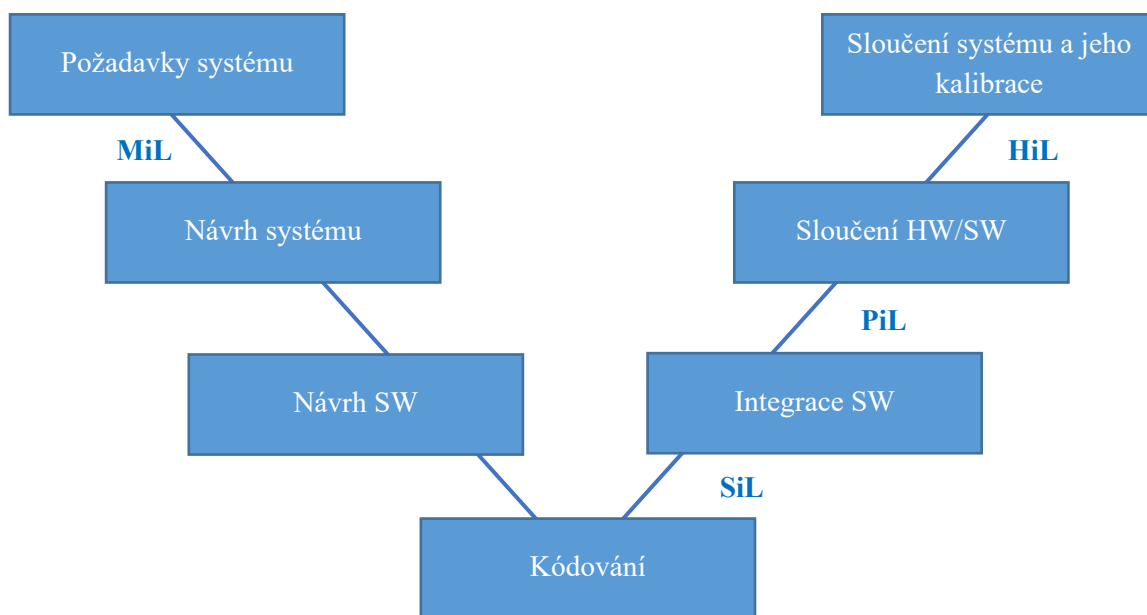
Pojem kosimulace neboli také psáno jako ko-simulace, je komplexnější obdoba simulace. Jedná se o techniku simulace, která vznikne propojením více simulátorů, přičemž každý simulátor je

zodpovědný za simulaci jen dané části systému. Všechny simulátory mohou běžet současně na jednom PC, nebo mohou být rozděleny na více PC. Výsledkem je tedy simulace celého systému, často vyobrazeného jako „black box“, s použitím více simulátorů. [4]

### 1.3 Simulační techniky

Pro ověření správnosti našeho postupu a verifikaci splnění zadaných požadavků, můžeme využít X-in-the-Loop techniky k testování aplikace v jednotlivých fázích jejího vývoje. Kvůli stádiu ověřování a schvalování došlo k lehké úpravě tradičního vývojového procesu vodopád a byl vytvořen V-model, který propojuje aktivity v počátečním vývoji s odpovídajícími aktivitami v pozdějším testování. V-model se může objevit v několika variantách, podle potřeby vyhovující pro jeho využití. [5; 6]

V-model je primitivní obměna tradičního vodopádového modelu systému nebo SW vývoje. V-model má větší důraz na ověřování a schvalování. V-model ohýbá směrem nahoru dolní polovinu vodopádového modelu, čímž vzniká tvar písmene V. Činnosti na pravé straně ověřují nebo schvalují činnosti nacházející se na levé straně. Levá strana zastupuje analýzu, která rozděluje požadavky na malé zvládnutelné kousky, zatímco pravá strana ukazuje syntézu, která slučuje kousky do systému, který odpovídá vyžadovaným potřebám uživatele. [6]



Obrázek 1.2: V model [7]

#### 1.3.1 Model-in-the-Loop (MiL)

MiL je označení pro techniku, kde je model určité funkce kontrolován se simulovaným modelem systému. Simulace probíhá tak, že matematický model i řídicí systém jsou simulovány v simulačním prostředí na stejném počítači. Tato technika se používá k ověření a optimalizaci konstrukce stroje nebo technologie, k definování vstupních a výstupních signálů a pro návrh struktury řízení. MiL testy jsou prováděny za pomoci vývojového prostředí MATLAB/Simulink. Testovací prostředí z MiL může být následně použito pro SiL testy. [5; 8]

### 1.3.2 Software-in-the-Loop (SiL)

SiL je ověřovací technika, kde je v simulačním (software) prostředí testován spustitelný kód, který může být vygenerován přímo z modelu. Je definován jako začlenění zkompilevaného software kódu do simulačního modelu. Matematický model je simulován v reálném čase společně s řídicím systémem na stejném počítači. SiL slouží k ověření výpočetní náročnosti řízení s danou periodou vzorkování a také pomáhá zvolit správný výkon řídicího hardwaru. [5; 9; 10]

### 1.3.3 Processor-in-the-Loop (PiL)

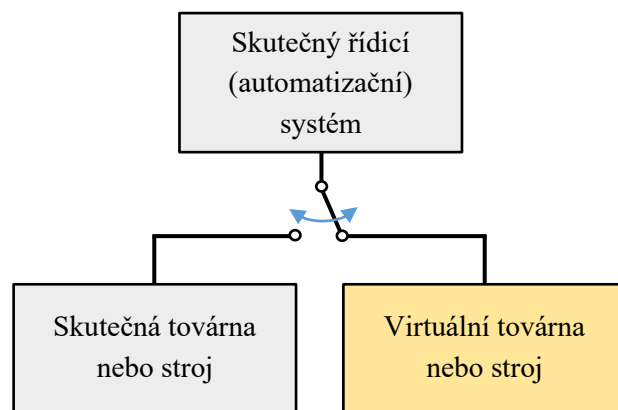
V technice PiL je procesor využíván pro testování spustitelného kódu. Matematický model je simulován v reálném čase. Řídicí systém je realizován na cílové HW platformě, procesor je připevněn na konkrétní hardware desce. Nejsou zde použity karty vstupů/výstupů, čidla a akční členy. Přenos dat mezi modelem a řídicím systémem je prováděn pomocí průmyslové komunikace. Využívá se pro ověření výpočetního výkonu řídicího HW, pro testování kritických situací atd. PiL vyplňuje mezeru mezi SiL a HiL. [5; 11]

### 1.3.4 Hardware-in-the-Loop (HiL)

Pojem HiL označuje standardní metodu pro vývoj a testování řídicích aplikací před jejich uvedením do provozu. Části skutečného systému jsou nahrazeny simulací. Spustitelný kód je testován na cílovém hardware, zatímco jeho prostředí je simulováno. Reálné a simulované části systému jsou propojeny a uspořádaný pro simulaci a testování dynamického chování skutečných komponent. HiL metoda je podobná jako PiL, simulace řídicího systému běží na cílové HW platformě. Navíc jsou použity karty vstupů/výstupů a model obsahuje emulátory senzorů a akčních členů. Jedná se o poslední krok před vývojem konečného prototypu. [5; 12]

## 1.4 Virtuální zprovoznění

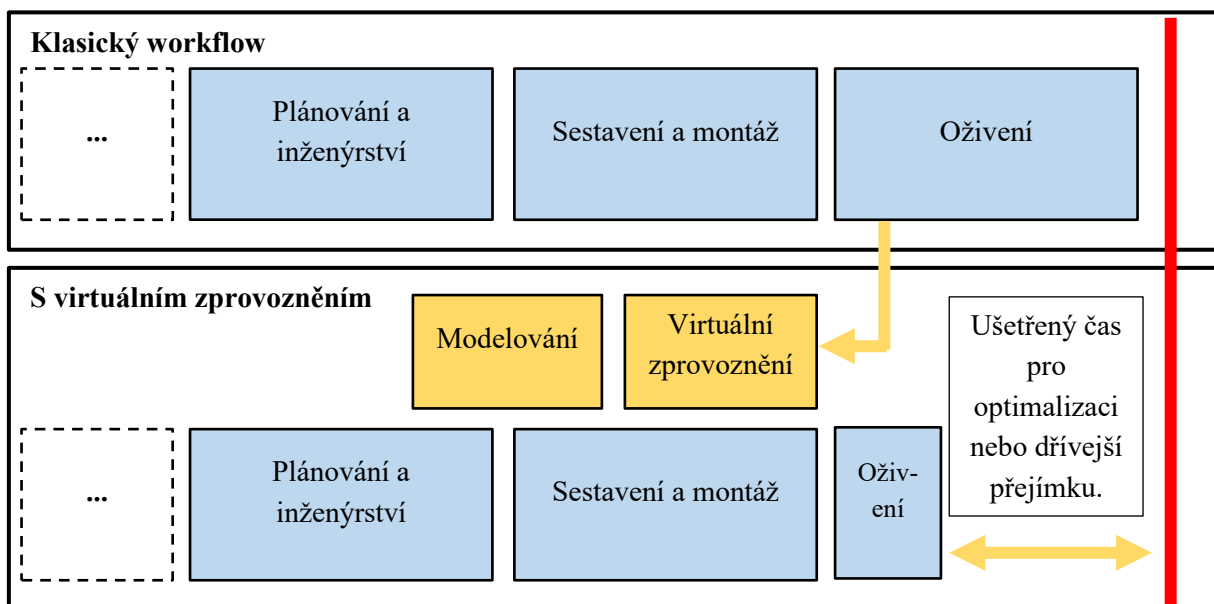
Virtuální zprovoznění, virtuální uvedení do provozu neboli virtuální oživení označuje jednu a tu stejnou činnost. Slovo „virtuální“ znamená, že se jedná o simulaci odehrávající na počítači nebo v počítačové síti. Pod „zprovozněním“ si můžeme představit projektování, instalaci, testování, provoz a údržbu všech systémů průmyslových zařízení. Spojením těchto dvou výrazů „virtuální zprovoznění“, vznikne proces využívající simulaci k testování strojního zařízení, robotické buňky nebo celé továrny. [13; 14]



Obrázek 1.3: Virtuální a klasické zprovoznění [15]

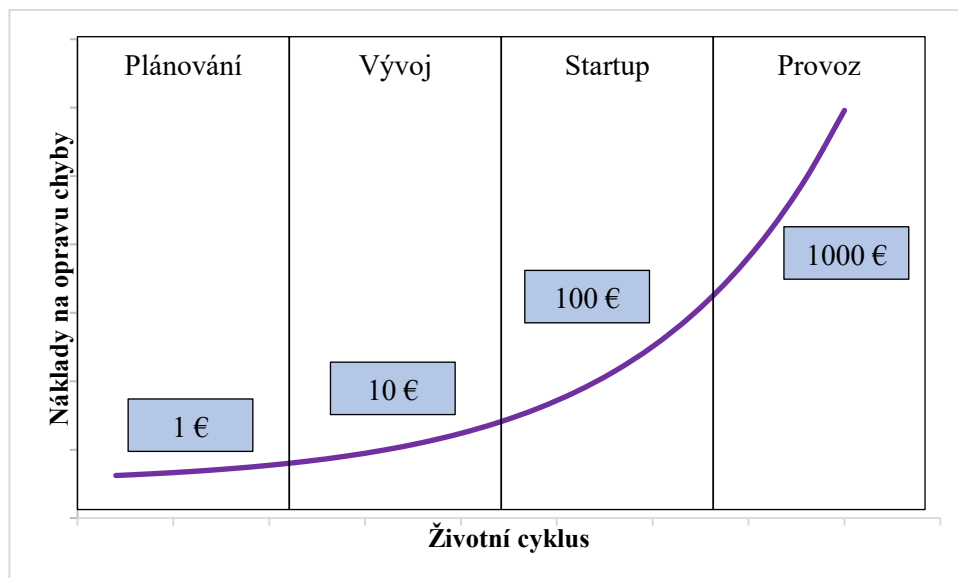
Simulační model systému je vytvořen, aby nahradil skutečný fyzický model a poskytuje virtuální prostředí pro ověření řídicího SW před samotnou instalací nebo výrobou stroje nebo celé továrny. Virtuálně může být oživen celý systém nebo jen jeho část např. buňka nebo stroj na lince. Simulace reaguje na výstupy z řídicího systému, na základě, kterých do něj zpět posílá simulované vstupy, tak jako v reálném světě. Můžeme ověřit SW pro zařízení jako je např. PLC, HMI, CNC nebo robotický kontrolér řídicí průmyslového robota. Simulace probíhá v pohodlí kanceláře bez nutnosti všeho potřebného HW. V některých případech je simulace vykonávána na digitálním dvojčeti. [14; 16]

**Přejímka**



Obrázek 1.4: Porovnání workflow při použití virtuálního zprovoznění [16]

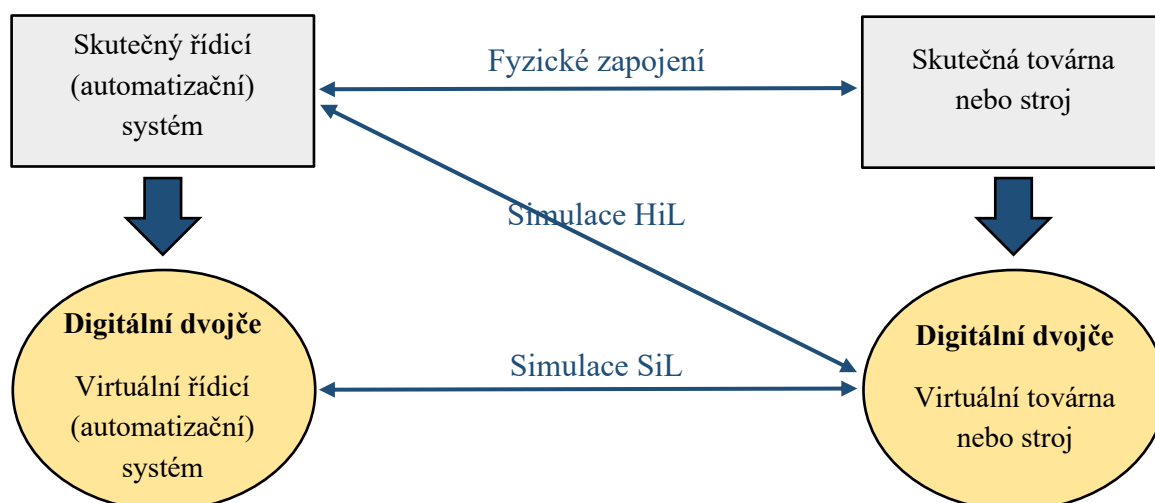
Použitím virtuálního zprovoznění dokážeme přesunout část, jinak strávenou při oživování na stavbě, do dřívější fáze projektu. Dojde ke snížení doby programátora na stavbě, která by měla být nižší až o 70 %. V rané fázi dojde k ověření SW a k efektivnějšímu engineeringu, vzhledem k včasné spolupráci mechanika, robotika a PLC programátora. Pomocí zprovoznění na PC můžeme testovat různé scénáře bez jakéhokoliv rizika, které by se mohlo promítnout do celkových nákladů projektu. Možné chyby by měly být eliminovány v co nejdřívější fázi projektu, jelikož náklady na jejich opravu s vývojem projektu exponenciálně rostou. Je udáváno, že s každým vývojovým stupněm rostou náklady až desetinásobně „Pravidlo desetinásobku“. [14; 16; 17; 18]



Obrázek 1.5: Desetinásobný nárůst nákladů s každou fází projektu [17]

## 1.5 Digitální dvojče

V dnešní době neustálého vývoje informačních technologií nové generace (New IT), do nichž patří cloud computing, internet věcí, big data, umělá inteligence atd., je postavení virtuálního prostoru stále více relevantní. Interakce mezi fyzickým a virtuálním prostorem je stále používanější. Architektura sjednocující oba světy je nazývána jako kyberneticko-fyzický systém (CPS). CPS si klade za cíl začlenit komunikační a výpočetní kapacity do fyzických aktiv, aby bylo možné fyzický prostor sledovat, koordinovat a řídit virtuální stranu v reálném čase. Protože digitální dvojče je cílená aplikace CPS a poskytuje praktičtější hodnoty a podrobnosti implementace, je představována jako pragmatický způsob bezproblémové integrace a splynutí. [19]



Obrázek 1.6: Digitální dvojče automatizačního systému a továrny [15]

Digitální dvojče lze zjednodušeně popsat jako velmi složitý virtuální model, který je přesným odrazem svého fyzického protějšku, během jeho celého životního cyklu. Dvojče pomáhá pochopit nejen, jak se produkt chová, ale i jak se bude chovat v budoucnu za pomoci simulace, odhadu a optimalizace

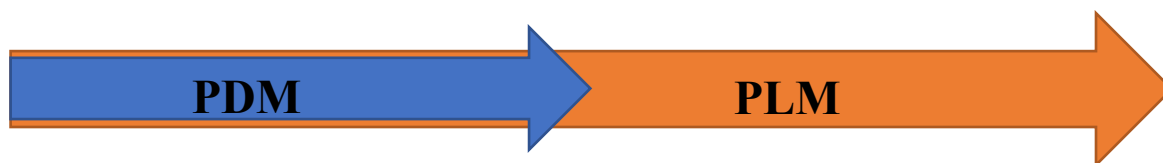
produktu nebo výrobního systému. Pro zajištění přesného modelování a zobrazení rozhodujících informací digitálního dvojčete, po celý životní cyklus produktu nebo jeho výroby, jsou použita data ze senzorů na fyzickém produktu, určující chování, provozní stav a změny v čase. Dvojčetem může být např. auto, letadlo, tunel, most, tryskový motor, celé město, továrna, jednoúčelový stroj nebo také řídicí systém. [20; 21]

### 1.5.1 Digitální dvojče a virtuální prototyp

Digitální dvojče úzce souvisí s virtuálním prototypem, což je statický 3D virtuální model, který nahrazuje fyzický prototyp do testovací a posuzovací fáze produktu. Virtuální model je vytvořen pomocí počítače metodou CAD nebo CAE. Model může být použit pro odhalení chyb a k odhadu chování konečného výrobku v brzké části vývoje. Virtuální prototyp lze považovat za základ digitálního dvojčete, nicméně nejedná se o stejný pojem. [19]

Digitální dvojče má oproti virtuálnímu prototypu řadu výhod. Virtuální prototyp je využíván hlavně ve fázi návrhu pro vyhodnocení a schválení produktu, zatímco virtuální model digitálního dvojčete doprovází svůj fyzický protějšek během celého životního cyklu, od výroby až po jeho likvidaci. Protože digitální dvojče sbírá hojně množství dat v různých stádiích (výroba, prodej, údržba), může brát v potaz vše co se může odehrávat ve fyzickém světě a vést k obsáhlejší úpravám ve fázi návrhu vedoucí k odstranění potenciálních poruch. Navíc díky datům z celého životního cyklu produktu má digitální dvojče schopnost motivovat k inovacím v designu. Virtuální prototyp není tolik propojen s fyzickým produktem, kdežto virtuální model digitálního dvojčete zůstává ve spojení během celého životního cyklu, aby odrážel skutečné stavy a základní poznatky v reálném čase, což může konstruktérům poskytnout cenné informace vedoucí k včasnému vylepšení produktu a k rychlému přizpůsobení trhu. Virtuální prototyp představuje pouze ideální produkt, kdežto digitální dvojče může představovat ideální i skutečný produkt. Model ideálního produktu je zkonstruován ve fázi návrhu a model skutečného produktu je vytvářen postupně. Po návrhu dochází k integraci skutečných údajů o produktu generovaných při výrobě, provozu, údržbě, likvidaci atd. Vzájemným porovnáním těchto dvou modelů lze snadno zjistit a eliminovat jejich rozdíly. [19]

## 1.6 PDM



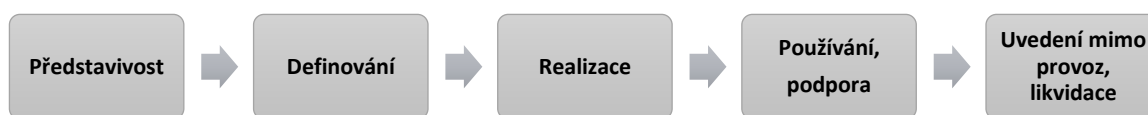
Obrázek 1.7: *PDM a PLM* [22]

S nárůstem využití CAD se začala data konstruktérů a projektantů stěhovat z papírů do CAD souborů. Protože počet souborů rychle narůstal a udržet přehlednost a pořádek bylo stále obtížnější, objevily se systémy PDM. PDM je systém, který je určen ke správě dat o produktech a s nimi spojených procesů. Jedná se o data jako jsou CAD modely, výkresy, kusovníky, údaje o dílech, specifikace produktu, NC programy, výsledky analýz atd. PDM se také stará o verzování jednotlivých dokumentů. [23]

## 1.7 PLM

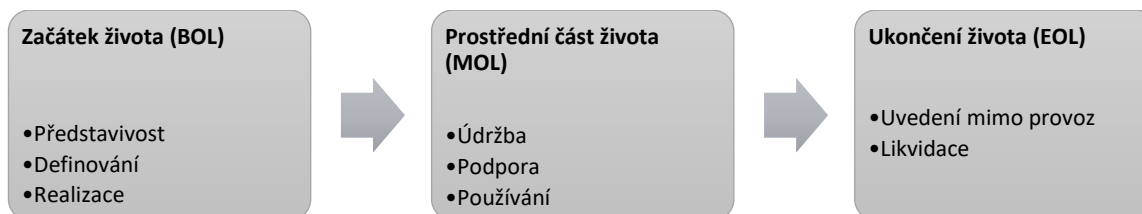
PLM je vyšší vývojový stupeň PDM. Zatímco PDM se zabývá pouze dat týkajícími se konstrukce výrobku, PLM spočívá v co nejefektivnější správě výrobku po jeho celý jeho životní cyklus. PLM spravuje produkty od prvního nápadu, včetně vývoje, přes růst, až k jeho likvidaci. Cílem je zvýšit zisk, snížit náklady spojené s výrobkem, maximalizovat hodnotu produktového portfolia a maximalizovat hodnotu, nejen současných, ale i budoucích výrobků pro zákazníky i akcionáře. [23; 24]

Životní cyklus výrobku má pět fází. V každé z pěti fází je v jiném stavu. Během fáze představivosti je výrobek pouhým nápadem v hlavě. Ve fázi definování jsou myšlenky převedeny do podrobného popisu. Na konci realizační fáze existuje výrobek ve své konečné podobě např. jako auto. Během fáze používání/podpora, je výrobek se zákazníkem, který jej používá. Nakonec se výrobek dostane do fáze, ve které již není užitečný, je uveden mimo provoz a je zlikvidován zákazníkem. [24]



Obrázek 1.8: *Pět fází životního cyklu výrobku* [24]

Výrobek musí být ve všech těchto fázích spravován, aby bylo zajištěno, že vše dobře funguje a výrobek je přínosem. Tři fáze životního cyklu (představivost, definice a realizace) tvoří začátek života výrobku (BOL). Prostřední část života výrobku (MOL) zahrnuje činnosti, jako je používání produktu, podpora a údržba. Poslední je část ukončení života (EOL), která zahrnuje činnosti, jako je uvedení výrobku mimo provoz, likvidace a recyklace. [24]



Obrázek 1.9: *BOL, MOL a EOL* [24]



## 2. Simulace v automatizaci



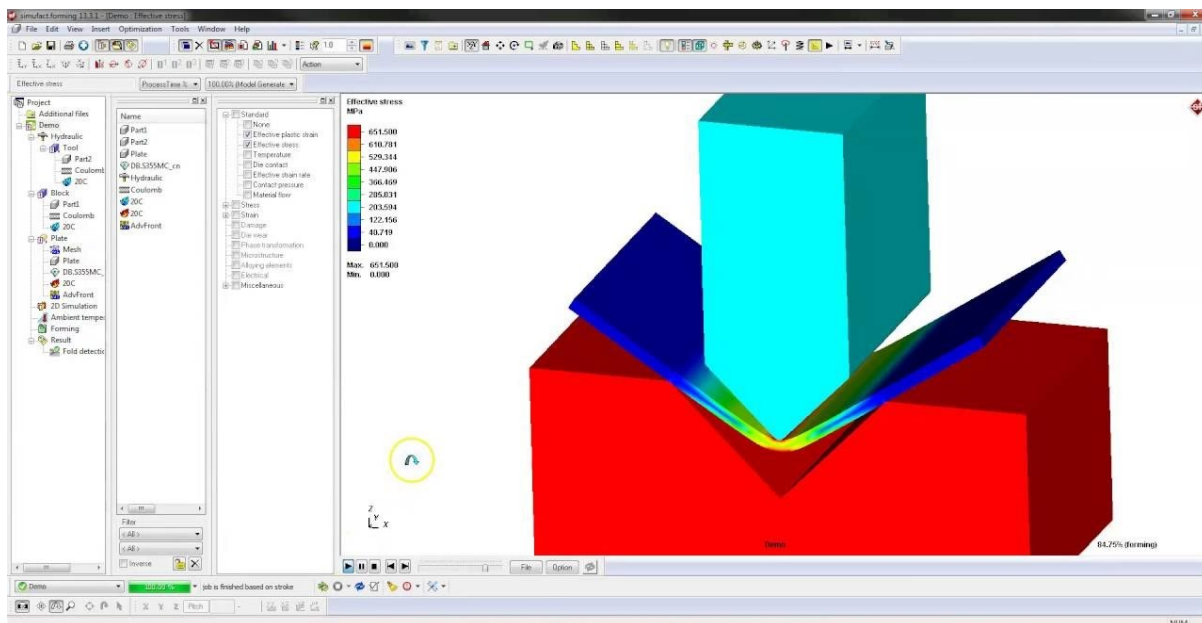
Obrázek 2.1: *Simulace výrobní továrny v automotive průmyslu [25]*

Průmyslové systémy jsou jednou z nejvíce používaných oblastí využívající simulaci. V roce 1960 zde byla simulace použita zcela poprvé. Simulace je v průmyslu používána při návrhu nových výrobních zařízení, skladů a distribučních center. Využívá se také pro vyhodnocení nově navržených úprav na stávajícím systému. Používáním simulace bylo zjištěno, že je podstatná pro posouzení celkových finančních investic a také do výsledného návrhu zařízení. Simulace je pro manažery užitečná pro poskytnutí tzn. „zkusební jízdy“ před provedením investice, aniž by došlo k narušení stávajícího systému. [2; 26]

Za posledních 20 let podíl SW na vývoji narostl až na 45 %. Objevují se stále komplexnější produkty. Na trhu narůstá globální konkurence, objevují se noví hráči a s nimi i nové trhy. Objevuje se také nárůst regulace v oblasti bezpečnosti provozu, spotřeby, ekonomiky a efektivity provozu zařízení. Zákazníci vyžadují čím dál tím vyšší rychlost dodávky stroje od jeho objednávky, flexibilitu stroje, schopného reagovat na budoucí změnu jejich výrobku, vyšší kvalitu a chtějí reagovat na ekonomické trendy, jako je efektivita, úspory při pořízení a provozu zařízení a ekologičnost celého provozu. Z těchto údajů lze jednoduše usoudit, že použití simulace v automatizaci bude v budoucnu jen narůstat. [27]

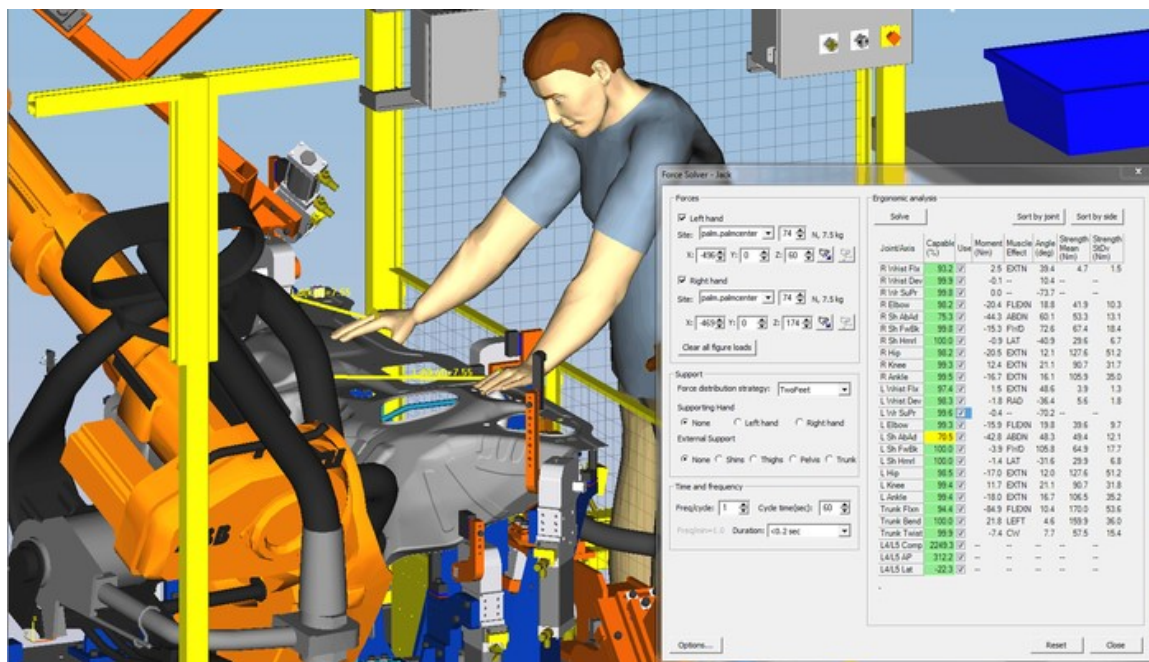
### 2.1 Možnosti

V průmyslové automatizaci může být simulováno několik různých typů zařízení nebo strojů. Za účelem ověření a optimalizace řídicí aplikace ve virtuálním prostředí je možné simulovat konkrétní zařízení např. PLC, DCS, IPC, CNC, HMI nebo robotický kontrolér včetně teach pendantu robota. V určitých případech může být tato simulace nedostačující nebo nepotřebná např. pokud vlastníme fyzický řídicí systém a simulovat chceme pouze stroj. Simulací stroje se simulace stává obtížnější. Čím více částí systému bude simulováno, tím více se simulovaný systém bude podobat skutečnému systému. Pomocí různých nástrojů, lze vytvořit model systému a zkoumat jeho dynamické chování nebo simulovat konkrétní technologický proces např. nýtování, utahování šroubů, svařování, obrábění, ohýbání, lisování, válcování nebo jiná tváření materiálu. V průmyslové automatizaci můžeme také simulovat dráhy robotů a ověřit jejich dosah, zjistit případné kolize nebo dokonce simulovat pohyb vedení kabelů a hadic podél ramen robota a ty následně optimalizovat.



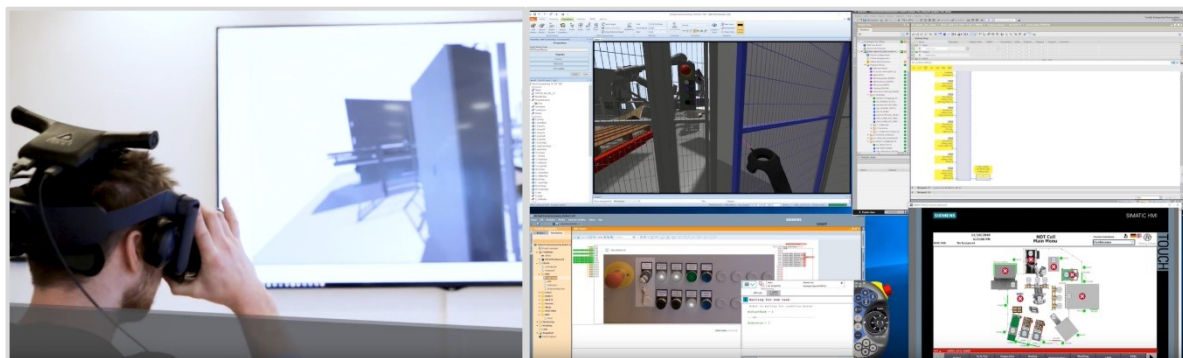
Obrázek 2.2: Simulace ohýbání plechu v prostředí Simufact Forming [28]

V automatizaci se nesimulují pouze zařízení, stroje a technologické procesy. V případě simulace virtuálního modelu celé linky nebo továrny jsou otevřeny možnosti ke zkoumání materiálového toku celým systémem. Takto lze snadno objevit slabá místa jako jsou nepřehledné a nadbytečné materiálové toky, neefektivní pohyby pracovníků, nevyužitá výrobní plocha atd. Za účelem optimalizace efektivity a rozložení pracovišť, kde pracují operátoři např. vkládání materiálu nebo montáž, slouží simulace člověkem prováděných úkonů. Ergonomické simulace dokáží zjistit časovou náročnost této práce, výkonnost, bezpečnost práce a sledovat dopad možných změn na člověka.



Obrázek 2.3: Simulace pracoviště s člověkem v prostředí Tecnomatix Process Simulate [29]

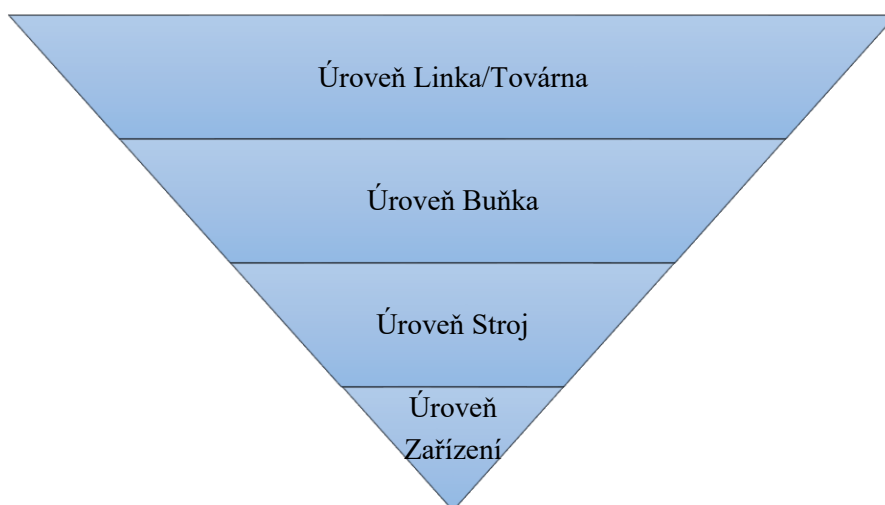
Zajímavým doplňkem během virtuálního zprovoznění stroje je možnost využití prvků virtuální reality. Kdokoliv s brýlemi a ovladačem pro VR si tak může celý proces výroby ve virtuálním světě projít, detailně prohlédnout a interagovat s virtuálním systémem během testů, tak jako ve skutečném světě. Lze snadno ověřit bezpečnostní funkce, generování alarmů a chování celého systému v různých situacích. VR je také možné použít pro zaškolení operátorů nebo údržbu. [30]



Obrázek 2.4: *Integrace VR do procesu virtuálního zprovoznění* [30]

## 2.2 Úrovně simulace

Simulace v průmyslové automatizaci za účelem virtuálního zprovoznění, může být rozdělena na několik úrovní podle komplexnosti simulovaného systému.



Obrázek 2.5: *Rozdělení úrovně simulace podle komplexnosti systému* [27; 18]

Úroveň zařízení – v této úrovni je simulován řídicí systém PLC, DCS, IPC nebo CNC, HMI panel, frekvenční měnič, motor, pneumatický nebo hydraulický válec a ostatní zařízení nacházející se v oblasti „field level“. Simulací v této úrovni je ověřen program, získána časová odezva a takt systému, otestována komunikace, vyzkoušena reakce programu na různé scénáře, ověřeno fungování bezpečnostní ochrany, je zjištěno, zda je elektrický pohon dostatečně výkonný, jaký může být dosažen maximální tlak, zda jsou zde rizika vibrací a jiné fyzikální vlastnosti zařízení. [27]

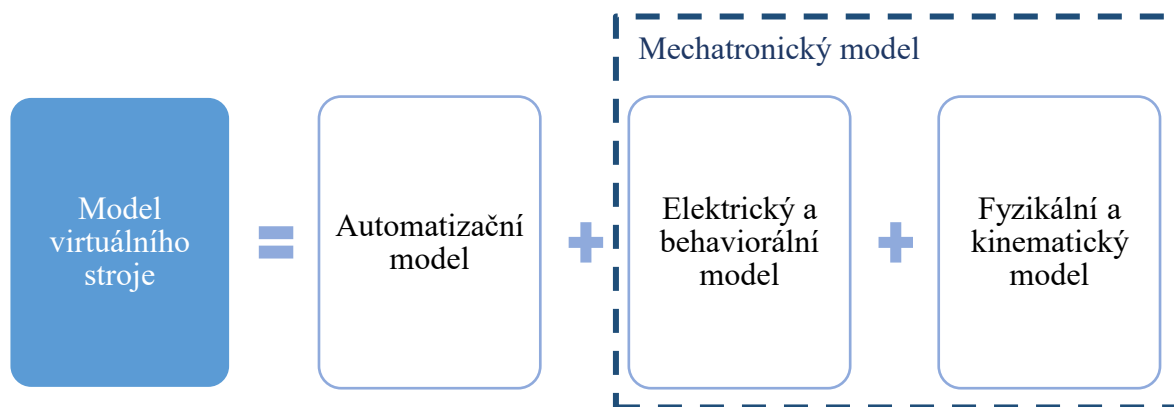
Úroveň stroje – v této úrovni je řešena simulace mechatronického zařízení nebo stroje. Ověřuje se zde chování mezi řídicím systémem a strojem. V simulaci se mohou také simulovat zařízení z nižší úrovně. Simulací zjistíme, zda naprogramovaný SW funguje dle očekávání, ověřuje se fungování bezpečnostních prvků, zjistíme, zda se ve stroji objevují kolize, jaký je takt stroje a zda se celkový návrh stroje shoduje s představou zákazníka. [27]

Úroveň buňky – v této úrovni můžeme simulovat celou robotickou buňku, včetně všech jejích součástí. Jedná se o simulaci průmyslových robotů a jejich řídicích kontrolérů. V simulaci se také mohou objevit zařízení z nižších úrovní, které zde mohou být také simulovány. Simulací zjistíme, zda je návrh layoutu správně, roboti dosáhnou do všech požadovaných pozic bez kolizí, výrobní takt pracoviště, chování pracoviště při různých situacích, ověření bezpečnosti celého pracoviště a programu robota včetně komunikace mezi robotickým kontrolérem a řídicím systémem. [27; 18]

Úroveň linky/továrny – v této úrovni je řešena simulace výrobní linky nebo dokonce celé továrny. V simulaci se objevují zařízení nižších úrovní, které mohou být také simulovány. Simulací v této úrovni zjistíme, průchodnost výrobní linky, nalezneme úzká místa, kde dochází ke zpomalení toku materiálu, zjistíme, jak velké mají být skladové zásoby, jak optimalizovat rozpracovanou výrobu, jaké strategie a řídicí parametry použít, jak naplánovat údržbu části linky a otestovat scénáře, jak je systém schopen reagovat na výpadky jednotlivých částí. [27; 18]

## 2.3 Model virtuálního stroje

Model virtuálního stroje je kombinací tří různých modelů. Automatizační model se stará o logiku řídicího programu, spadá zde např. PLC program a vizualizace. Automatizační model lze označit za simulační model pouze v případě, že je daný řídicí systém simulován. Následující dva modely jsou vždy simulační. Elektrický a behaviorální model se zabývá aktivními částmi systému jako jsou pohony, pneumatické nebo hydraulické ventily, chováním periferií atd. Fyzikální a kinematický model řeší mechanické části systému a stará se o vizualizaci systému ve 3D. Model virtuálního stroje může také představovat digitální dvojče. [27]



Obrázek 2.6: Model virtuálního stroje [18; 27; 31]

V závislosti na strategiích firem vyvíjející simulační SW pro virtuální zprovoznění je možné se setkat i s odlišným modelem virtuálního stroje. Elektrický a behaviorální model a fyzikální a kinematický model je nahrazen tzv. mechatronickým modelem. Modely jsou integrovány do jednoho



společného modelu. Mechatronický model proto můžeme bez problému vytvořit v jednom vývojovém prostředí. [31]

### 3. Simulační platformy

Nezbytnou součástí pro virtuální zprovoznění jednoho nebo několika strojních zařízení nebo celé továrny je volba správné simulační platformy. Na trhu se objevuje řada různých druhů, lišící se, jak možnostmi, tak i cenou. V této kapitole jsou vybrané SW nástroje popsány a zařazeny podle modelu virtuálního stroje a úrovně simulace viz. tabulka níže. Buňky označené symbolem hvězdičky mohou být do dané sekce zařazený jen v některých případech.

Tabulka 3.1: *Zařazení vybraných nástrojů podle modelu virtuálního stroje a úrovně simulace*

Nástroj	Model virtuálního stroje			Úrovně simulace			
	Automatizační	Elektrický a behaviorální	Fyzikální a kinematický	Zařízení	Stroj	Buňka	Linka/Továrna
PLC simulátory	X	X*		X			
HMI simulátory	X			X			
Robotické simulátory	X		X	X		X*	
SIMIT		X		X	X	X	X
Simcenter Amesim		X	X	X	X		
MATLAB		X		X	X	X	X
FluidSIM		X		X			
NX Mechatronics Concept Designer			X	X	X		
Tecnomatix Process Simulate			X		X	X	
Tecnomatix Plant Simulate			X				X
WinMOD		X	X	X	X	X	X
Visual Components			X		X	X	X
ISG Virtuos		X	X			X	X
DELMIA V5			X		X	X	
CIROS Studio/Education			X		X		
Factory IO			X			X	

#### 3.1 PLC simulátory

Do této sekce spadají všechny simulátory PLC. Jejich použití se odvíjí zejména dle použitého vývojového prostředí a plánovaného reálného zařízení na skutečném systému. V modelu virtuálního stroje se jedná o automatizační simulační model. Na virtuálně zprovozněném PLC dokážeme otestovat a optimalizovat řídicí SW nebo otestovat komunikaci mezi dvěma a více PLC. Bez použití dalších simulačních modelů je simulace značně omezená, jelikož vstupní signály je zapotřebí ve vývojovém

prostředí manuálně nahazovat. Simulátor PLC může také zastřešit elektrický a behaviorální model vytvořením druhé instance PLC. Simulační nástroje různých výrobců PLC jsou vypsány v tabulce níže.

Tabulka 3.2: *Seznam simulátoru PLC*

Výrobce PLC	Nástroje
ABB	Automation Builder (simulátor PLC je součástí vývojového prostředí)
Allen-Bradley	RSLogix Emulator 5000, Studio 5000 Logix Emulate
Beckhoff	TwinCAT (simulátor PLC je součástí vývojového prostředí)
Honeywell	SoftMaster-SIM
Mitsubishi	GX Simulator
Omron	CX-Simulator
Siemens	PLCSIM, PLCSIM Advanced
<i>Není výrobcem</i>	Codesys (simulátor PLC je součástí vývojového prostředí)

### 3.2 HMI simulátory

V této oblasti se nachází simulátory HMI. Jejich použití se odvíjí zejména dle výrobce plánovaného reálného zařízení. Možnost simulace HMI je nesmírnou výhodou při vývoji SW. Odpadá nutnost nahrávat SW do HMI po nález každé drobné chyby. Pomocí simulace jsme schopni tyto drobné chyby odhalit už před nahráním SW do fyzického panelu, což šetří čas. Simulátor HMI je také možné použít i na stávajícím fyzickém systému, kdy chceme technologii řídit a sledovat z jiného místa, než na kterém je umístěno fyzické HMI. Jednoduše to provedeme se spuštěným simulátorem HMI na notebooku. Komunikace pak probíhá mezi simulovaným HMI a fyzickým PLC. Simulační nástroje různých výrobců HMI jsou vypsány v tabulce níže.

Tabulka 3.3: *Seznam simulátoru HMI*

Výrobce HMI	Nástroje
ABB	PB610 Panel Builder 600: The HMI Simulator
Allen-Bradley	FactoryTalk View ME Station
Beckhoff	TwinCAT (simulátor HMI je součástí vývojového prostředí)
Mitsubishi	GT Simulator
Omron	CX-Simulator
Siemens	WinCC: HMI Runtime Simulation
<i>Není výrobcem</i>	Codesys (simulátor HMI je součástí vývojového prostředí)

### 3.3 Robotické simulátory

Do této kategorie spadají nástroje pro simulaci robotů pouze jednoho výrobce. Jsou to nástroje tvořené samotným výrobcem. Jejich hlavní nevýhodou je omezení pouze na jednu značku robotů. V prostředí je většinou emulován kontrolér i teach pendant a je zde umožněno vytvářet program robota v dostupném programovacím jazyce. Pro ABB roboty je to jazyk RAPID, pro roboty Stäubli zase jazyk VAL3. Nástroje mohou podporovat kosimulaci, a tak robotický SW může pokrývat jen automatizační

model virtuálního stroje a fyzikální a kinematický model může zastávat jiný simulační nástroj. Můžeme také nechat robotický SW pokrýt oba modely virtuálního stroje. RobotStudio může být například propojeno s reálným PLC pro simulaci akčních členů, které nejsou řízeny robotem (dopravník, lis, bezpečnostní dveře atd.).

Tabulka 3.4: *Seznam simulátorů robotů*

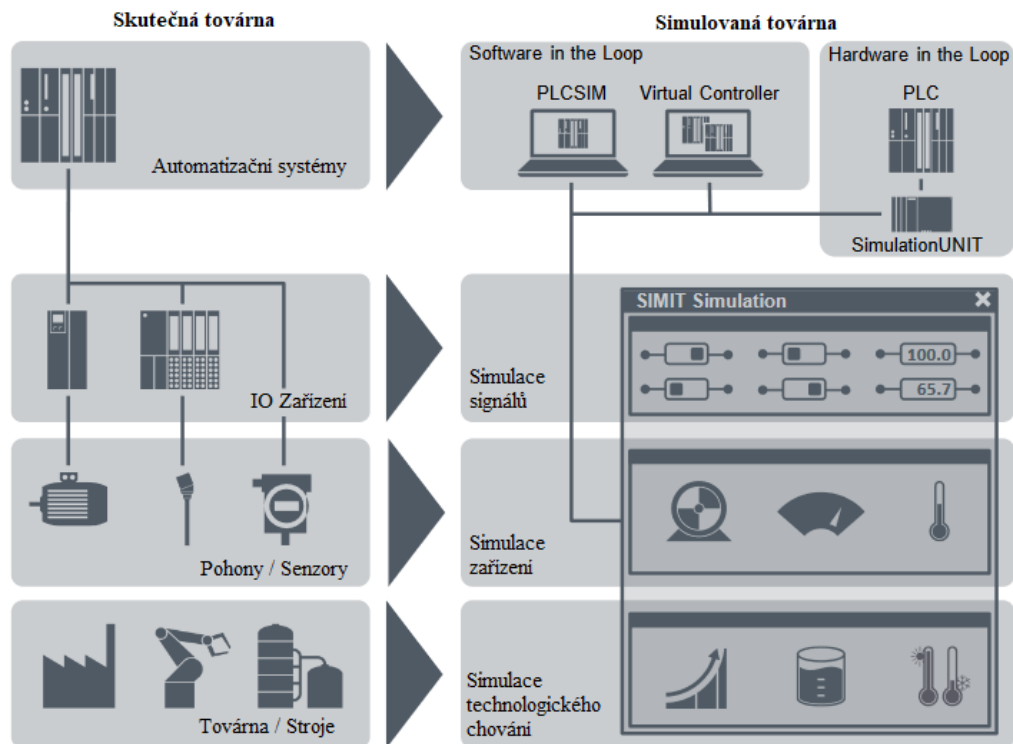
Výrobce robotů	Nástroje
ABB	RobotStudio
FANUC	ROBOGUIDE
Kawasaki	K-Roset
KUKA	KUKA.OfficeLite, KUKA.Sim
Mitsubishi	MELFA-Works
Stäubli	Stäubli Robotics Suite
Yaskawa	MotoSim EG

### 3.4 SIMIT

SIMIT je simulační software vyvíjený společností Siemens. Software vykonává simulaci v reálném čase a využívá přístupu systémové dynamiky, dynamika je v tomto případě čas. Uživatel může virtuálně spustit zařízení, stroje a procesy. Prostředí je také využíváno pro zaškolení budoucích operátorů továrny. Platformu SIMIT můžeme jednoduše používat bez dostupného vybavení nebo rozsáhlé technické znalosti modelování procesů. SIMIT nabízí moderní a uživatelsky přívětivé grafické rozhraní pro intuitivní ovládání. Automatizační projekty mohou být díky tomu realizovány včas, efektivně a spolehlivě. [32; 33]

SIMIT může být použit jako simulátor pro testování vstupních a výstupních signálů PLC, pro testování a uvedení do provozu automatizačního softwaru nebo také jako simulátor celé továrny. Z pohledu virtuálního modelu stroje SIMIT patří čistě do elektrického a behaviorálního modelu. Z počátku je možné použít SIMIT jen jako uživatelské rozhraní pro testování signálů a simulační modely můžeme přidat kdykoliv později, pro potřebu simulace chování továrny a provádění dynamických testů. [33; 34]



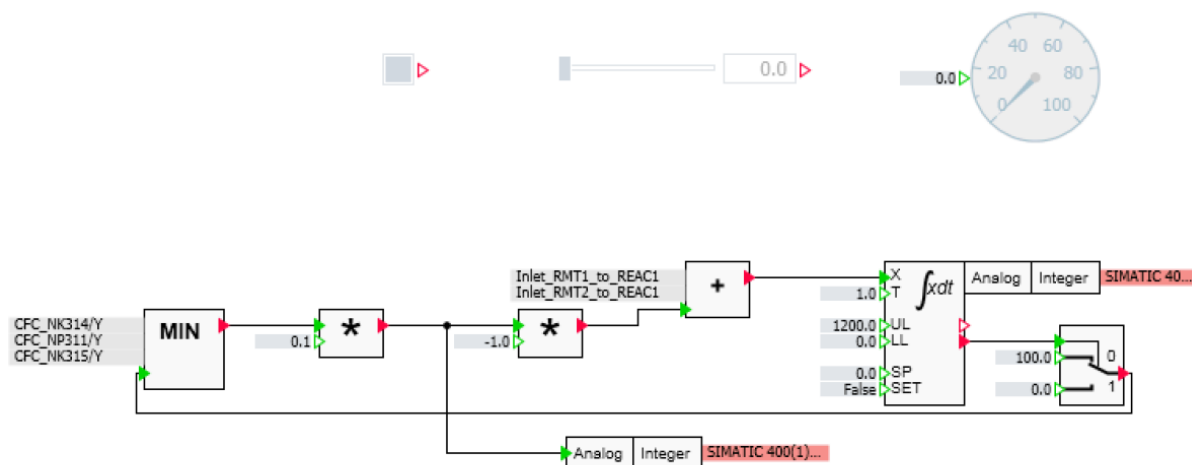


Obrázek 3.1: Porovnání skutečné a simulované továrny [34]

Simulace v SIMIT je rozdělena na tři základní úrovně. Jedná se o signální úroveň, úroveň zařízení a procesní úroveň. Do signální úrovně spadá testování a simulace signálů s fyzickým nebo simulovaným PLC. V úrovni zařízení je řešena simulace pohonů a senzorů v systému pomocí již vytvořených komponent pro simulaci chování těchto zařízení. Komponenty mohou být následně propojeny se signály ze signální úrovně. Technologické chování továrny a strojů je řešeno v procesní úrovni za pomoci komponent z rozšiřujících knihoven. [34]

SIMIT V10 je k dispozici ve verzích XS, S, M, L nebo XL. S rostoucí velikostí varianty roste maximální počet použitých simulačních tagů. Ve verzi XL neplatí žádná omezení, maximální počet simulačních tagů se odvíjí od výkonu PC. Balíček SIMIT obsahuje nástroje „Simulation Platform“ a „Virtual Controller“. Simulační platforma slouží pro tvorbu simulačních modelů. Pro emulaci řídicích systému řady S7-400 slouží virtuální řídicí systém. [33]

### 3.4.1 Struktura a elementy







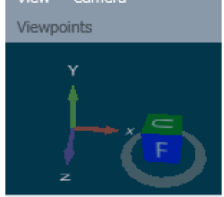




Obrázek 3.2: Příklad diagramu s propojenými komponenty a řídicími prvky [35]







Pro práci s nástrojem SIMIT je důležité znát následující elementy. *Diagram (Chart)* obsahuje simulační model. Komponenty, řídicí prvky a propojení mezi nimi je graficky znázorněno. Diagramy se vytváří a upravují v editoru diagramu. Simulační model může obsahovat více diagramů. *Komponenty (Components)* jsou dostupné v knihovnách a poskytují logické a aritmetické funkce, pohony, senzory, propojení mezi nimi a komunikaci. *Řídicí prvky (Controls)* slouží pro vkládání a zobrazování hodnot. *Vizualizace (Visualization)* umožňuje vytvářet diagramy s řídicími a zobrazovacími prvky, buďto manuálním vytvořením diagramů nebo automatickým vygenerováním diagramů z existujících dat. Grafické nástroje jako jsou čáry, obdélníky, elipsy atd. mohou být statické nebo animované, což umožňuje vytvářet komplexní dvourozměrné animace. Vizualizace poskytuje ideální přehled všech signálů v systému. Hodnoty, patřící do stejné části systému, mohou být rovněž uvedeny seskupeným způsobem, nezávisle na používaných fyzických adresách. *Propojení (Coupling)* definuje rozhraní, přes které bude SIMIT komunikovat s automatizovaným systémem. Mohou zde být importovány a upravovány signály různého formátu. [33]

Tabulka 3.5: Typy řídicích prvků [33]

Název	Datový typ	Použití	Symbol
Binární displej	Binární	Zobrazení signálu	False
Analogový displej	Analogový, integer	Zobrazení signálu	0.0
Digitální displej	Analogový, integer	Zobrazení signálu	0.0
Sloupcový graf	Analogový, integer	Zobrazení signálu	0 100
Tlačítko	Binární	Vstupní signál	
Tlačítko s obrázkem	Binární	Vstupní signál	

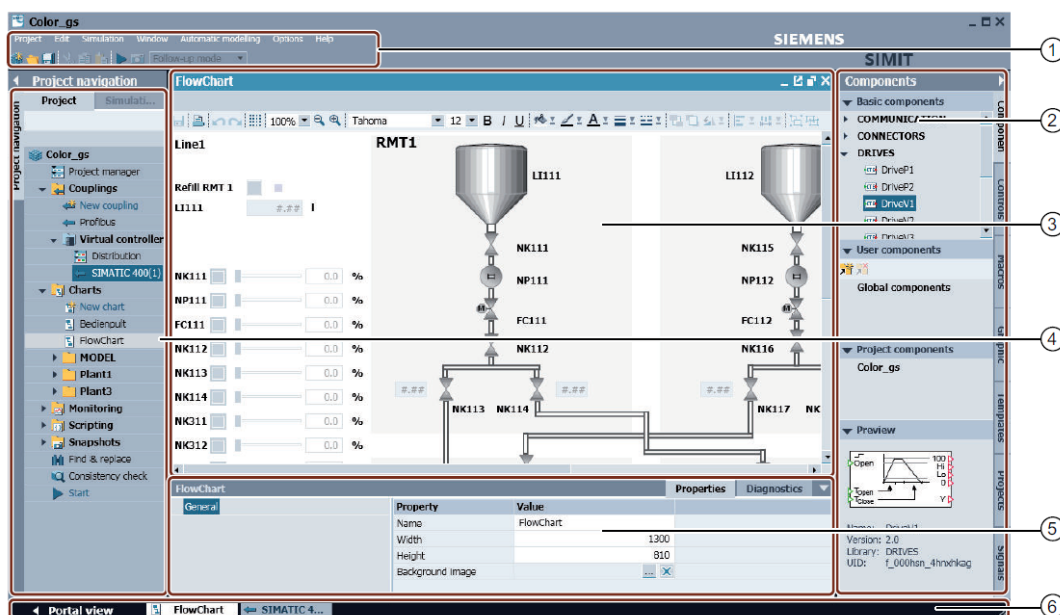
Přepínač	Binární	Vstupní signál	
Přepínač s obrázkem	Binární	Vstupní signál	
Stupňový přepínač	Integer	Vstupní signál	
Stupňový přepínač s obrázkem	Integer	Vstupní signál	
Digitální vstup	Analogový, integer	Vstupní signál	
Posuvník	Analogový	Vstupní signál	
3D prohlížeč	-	Integrace 3D grafiky do grafu	
Dělič signálu	-	Rozdělení signálů, musí být připojeny k jinému řídicímu prvku	
Akce	-	Otevírá grafy a trendy	

Tabulka 3.6: Typy grafických prvků [33]

Grafický prvek	Popis	Nastavení
 Text	Vloží text kdekoliv v diagramu	Písmo, velikost písma, barva písma, barva pozadí, barva ohraničení
 Line	Nakreslí rovnou čáru	Barva čáry, tloušťka čáry
 Rectangle	Nakreslí obdélník	Barva ohraničení, tloušťka ohraničení, barva výplně
 Ellipsis	Nakreslí elipsu	Barva ohraničení, tloušťka ohraničení, barva výplně
 Polyline	Nakreslí lomenou čáru	Barva čáry, tloušťka čáry
 Ellipse arc	Nakreslí oblouk elipsy	Barva čáry, tloušťka čáry
 Bezier curve	Nakreslí Bezierovu křivku	Barva čáry, tloušťka čáry

Součástí SIMIT jsou knihovny obsahující různé komponenty a řídicí prvky viz. Tabulka 3.5. V knihovně „Basic“ se nachází základní funkce pro vytváření simulací. Jsou zde konektory, bloky pro matematické a logické operace, analogové, integrační a konverzní funkce, komponenty pro simulaci ventilů, pohonů, motorů, senzorů a komunikace s různými zařízeními. V rozšiřujících knihovnách FLOWNET a CHEM-BASIC se nachází komponenty pro simulaci potrubních sítí. Knihovna CHEM-BASIC nalezne uplatnění při vytváření simulací v chemickém a farmaceutickém průmyslu. Knihovna CONTEC je také rozšířením, poskytující komponenty pro simulaci dopravníkových systémů ve 2D prostředí. [33]

### 3.4.2 Uživatelské prostředí

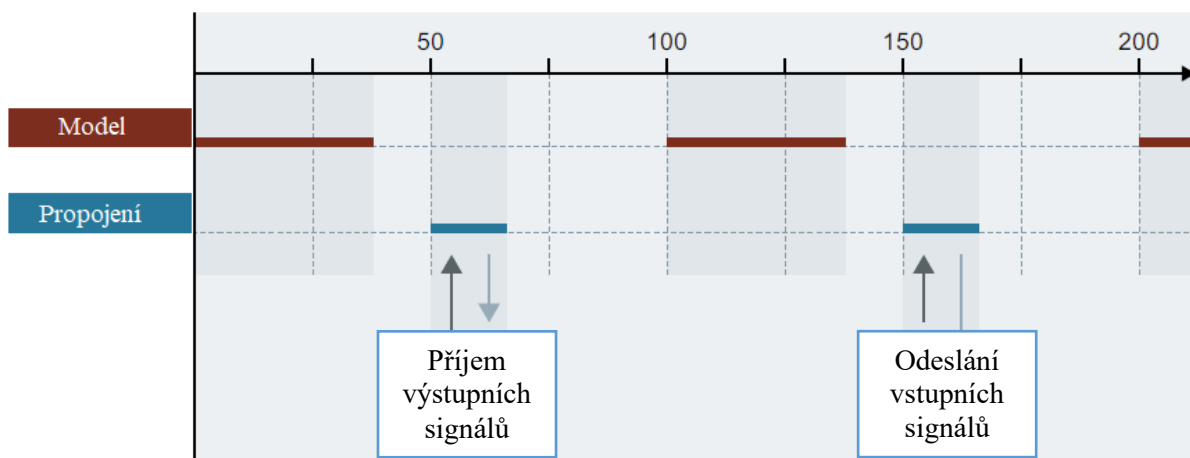


Obrázek 3.3: Popis uživatelského prostředí SIMIT [35]

① Menu lišta a panel nástrojů. Přístup k funkcím SIMIT. ② Karty úloh. Objekty jako knihovna komponent, řídicí a grafické objekty. ③ Pracovní plocha. Místo, kde probíhá editování. ④ Navigace v projektu. Zobrazení aktuálního projektu ve stromové struktuře. ⑤ Pohled na proměnnou. Jsou zde zobrazeny proměnné vybraného objektu. ⑥ Editační a statusová lišta. Zde lze přepínat mezi otevřenými editory a vzhledem portálu. Podle potřeby je zde zobrazena informace o aktuálním stavu SIMIT. [33]

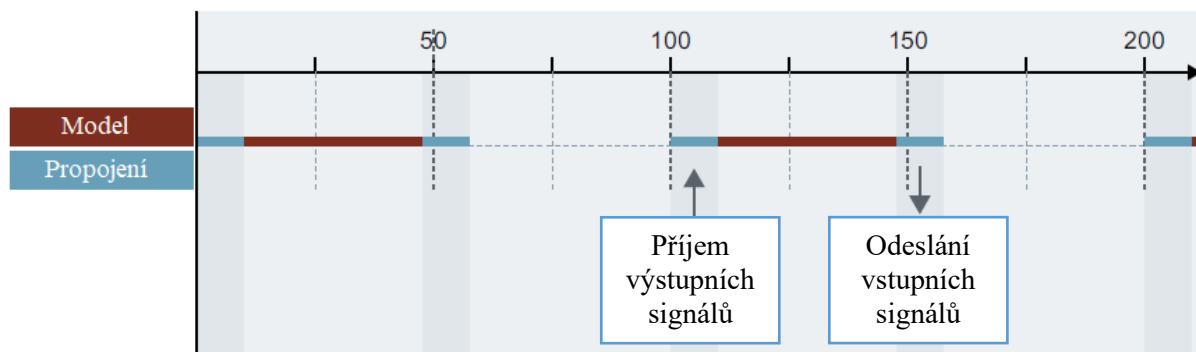
### 3.4.3 Pracovní módy simulace

Simulační model a propojení s jiným automatizovaným systémem může pracovat ve třech různých módech. Asynchronní, synchronní nebo pracovní mód synchronní sběrnice. Volba vhodného pracovního módu závisí na daném projektu. [33]



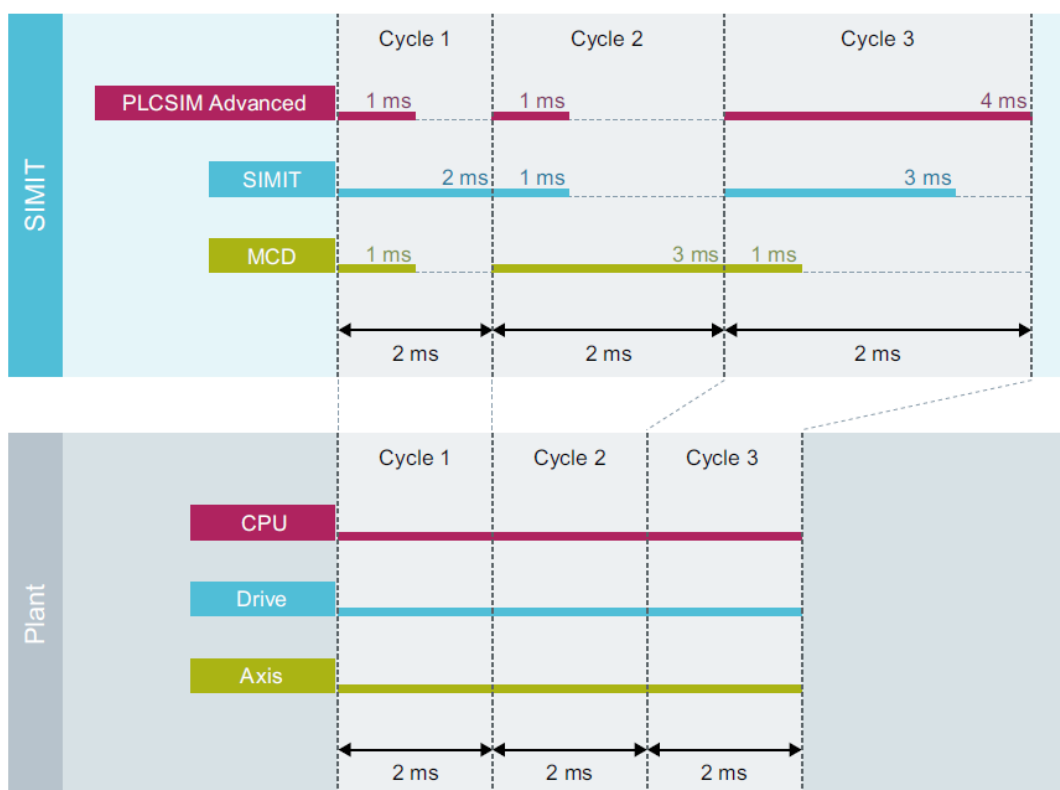
Obrázek 3.4: SIMIT – asynchronní pracovní mód [33]

U asynchronního pracovního módu jsou výpočty jednotlivých časových cyklů na modelu časově řízené. Pokud nedojde k dokončení běhu simulačního modelu v přiděleném čase, dojde k vynechání jednoho nebo i více cyklů. Výpočty jsou pak provedeny v dalších cyklech. To znamená, že v případě komunikačního problému není ovlivněn celý proces. Pro získání co nejrychlejší časové odezvy je zpracování signálů z propojení posunuto s časovým offsetem vůči kalkulaci modelu. [33]



Obrázek 3.5: *SIMIT – synchronní pracovní mód* [33]

V synchronním pracovním módu jsou modely a propojení kalkulovány v přesně danou postupnou sekvenci. Následující akce nenastane, dokud nebyla předchozí akce dokončena. Výsledkem je lepší časová odezva s kratšími doby jednotlivých cyklů. Nevýhodou je, že jakýkoliv model nebo propojení s automatizačním systémem dokáže zastavit vykonávání celé simulace. [33]



Obrázek 3.6: *SIMIT – pracovní mód synchronní sběrnice* [33]

S pracovním módem synchronní sběrnice SIMIT zajistí, že všechny komponenty zapojené do simulace mají totožný synchronizovaný simulační vývoj. Mód je především vhodný pro real-time aplikace. Pro aplikace bez real-time požadavku je vyhovující asynchronní nebo synchronní mód. Tento mód je dostupný pro propojení s automatizačními nástroji jako je PLCSIM Advanced a NX Mechatronics Designer. Cyklus je dokončen pouze pokud jsou dokončeny výpočty ve všech třech modelech. Jednotlivé nástroje uzpůsobí simulaci tak, aby byla co nejpodobnější realitě. Na grafu níže je zobrazeno časování CPU, měniče a osy ve skutečném světě. [33]

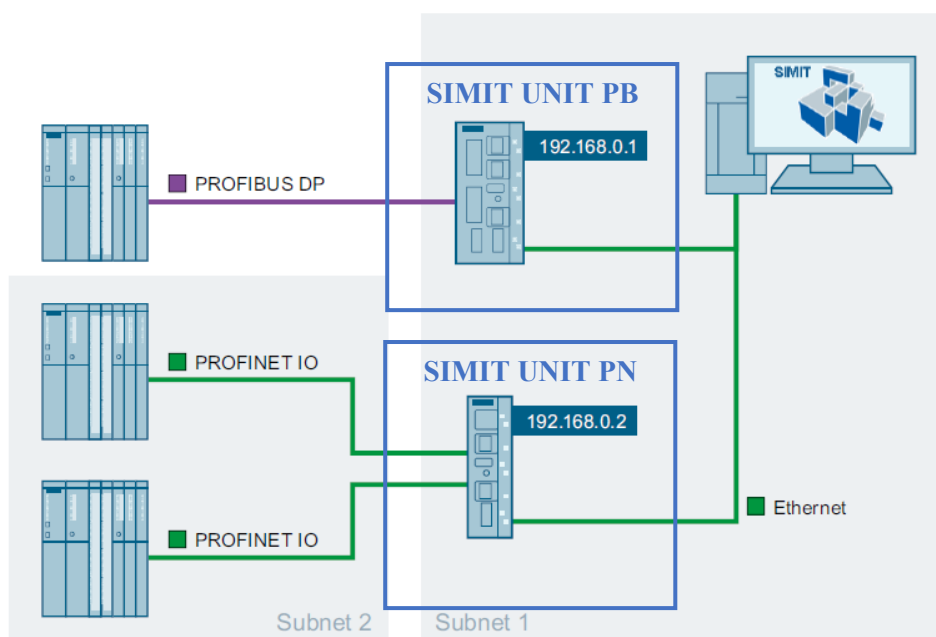
### 3.4.4 Typy propojení

SIMIT dovoluje tři různé typy propojení. Propojení s fyzickým SIMATIC řídicím systémem, propojení s emulovaným SIMATIC řídicím systémem a propojení s externím partnerem. Propojení SIMIT s PLC SIMATIC je možné s využitím přístupu simulace HiL, PiL nebo SiL. [33]

Tabulka 3.7: *Typy propojení se SIMIT* [33]

Propojení s fyzickým SIMATIC řídicím systémem (HiL a PiL)			Propojení s emulovaným SIMATIC řídicím systémem (SiL)			Propojení s externím partnerem	
SIMIT jednotka	PRODAVE	OPC UA	SIMIT VC	PLCSIM	PLCSIM Advanced	NX MCD	-

V případě HiL je propojení uskutečněno s fyzickým PLC jako je např. SIMATIC S7-400 přes PROFIBUS DP, PROFINET nebo PRODAVE. K úspěšnému provedení této simulace je zapotřebí vlastnit HW zařízení tzn. simulační jednotku SIMIT. Jednotka existuje ve dvou různých provedení v závislosti na typu komunikačního rozhraní (PROFINET, PROFIBUS DP) mezi PC a fyzickým PLC. [33; 36]



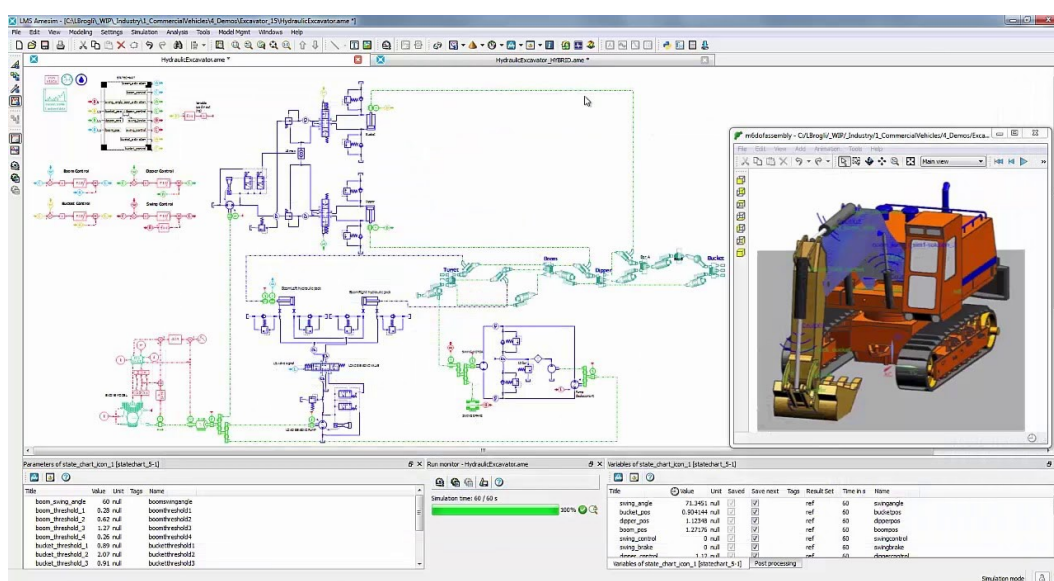
Obrázek 3.7: *Propojení SIMIT jednotky k SIMIT platformě* [33]

Nutnost použití simulační jednotky odpadá, pokud se smíříme se simulačním přístupem PiL. V tomto případě může komunikace mezi fyzickým PLC a SIMIT simulační platformou probíhat pomocí rozhraní OPC UA. K tomu je zapotřebí vytvořit OPC server, pomocí kterého mohou klienti na síti vyměňovat svá data.

U přístupu SiL je propojení provedeno s nástrojem PLCSIM, PLCSIM Advanced nebo SIMIT Virtual Controller. Pro použití SIMIT a PLCSIM nebo SIMIT a PLCSIM Advanced musí být oba nástroje nainstalovány na stejném PC. PLCSIM Advanced je rozsáhlejší verze klasického PLCSIM umožňující více možnosti simulace. Virtual Controller simuluje řídicí systém řady S7-300 nebo S7-400. Tyto emulované řídicí systémy mohou být distribuovány na několika PC. [33; 36]

### 3.5 Simcenter Amesim

Umožňuje posoudit a optimalizovat chování a výkon mechatronického systému během jeho vývoje nebo i za provozu, a to vše ve virtuálním světě za pomoci simulace. Obsahuje multifyzikální knihovny s tisícovkou modelů pro mechanické, elektrické, tepelné, regulační, hydraulické, pneumatické a chemické simulace. Nástroj Amesim obsahuje otevřené prostředí, které je snadno integrovatelné do různých podnikových procesů. Amesim umožňuje přesné modelování kinematického a dynamického chování, vedoucí k rychlé analýze velkého množství vlivů na chování celého systému. Ve virtuálním modelu stroje může Simcenter Amesim tedy pokrývat fyzikální a kinematický model společně s modelem elektrickým a behaviorálním, tento celek můžeme označit jako mechatronický model. [37; 38]



Obrázek 3.8: Vývojové prostředí Simcenter Amesim – simulace rypadla [39]

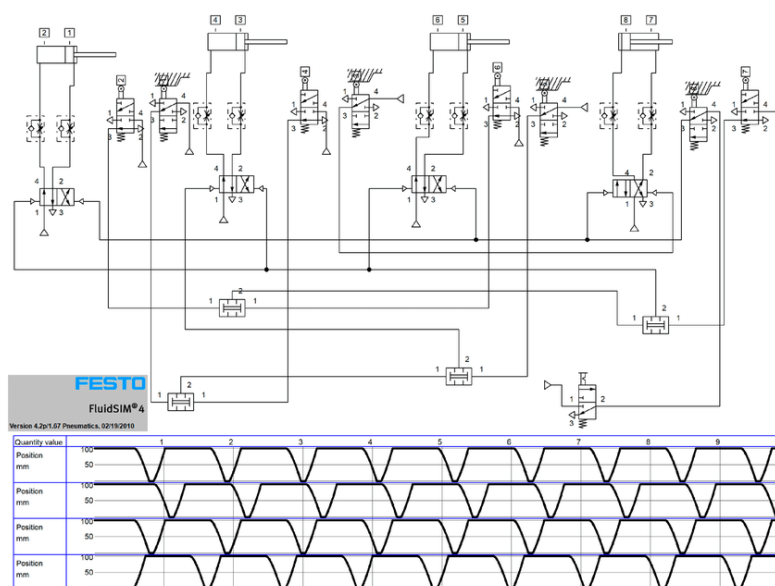
Vytvořené simulace lze snadno integrovat s CAE, CAD, SW pro řízení, FMI (Functional Mock-Up Interface), programem Modelica a s dalšími Simcenter nástroji (FloMaster, Star-CMM+, 3D, Motorsolve, FloTherm, FloEFD, Magnet, BDS, SCADAS, Sound Camera, Soundbrush, TestLab. Mezi možné aplikace v automobilovém průmyslu patří např. skutečný odhad emisí z jízdy, modelování ventilového rozvodu a systému mazání, návrh baterií a elektrických motorů, návrh palivových systémů, tepelný management, simulace výkonu baterií aj. [37; 38]



### 3.6 MATLAB

Prostředí MATLAB a Simulink obsahuje nástroje, které jsou velmi vhodné k modelování a simulaci dynamického chování fyzikálních systémů v různých oblastech průmyslu. Vytvořené modely jsou popsány rovnicemi nebo jsou sestaveny z bloků vyobrazující reálný systém. Tyto nástroje umožňují testovat kvalitu a robustnost navržených regulačních algoritmů. Příklady použití mohou být simulace pozice pístu v závislosti na změně přiváděného tlaku, simulace chování regulátoru, simulace ABS systému v automobilovém průmyslu nebo simulace strojového vidění s použitím deep learningu. [40]

### 3.7 FluidSIM



Obrázek 3.9: Vývojové prostředí FluidSIM – simulace 4 dvojčinných pneumatických válců [41]

FluidSIM slouží především pro simulaci kapalinových pohonů. Kromě pneumatických a hydraulických zařízení lze zde simulovat i elektronická zařízení. Tento nástroj umožňuje především kreslit elektro-pneumatické/hydraulické obvody na kterých lze následně vykonávat simulace. Při simulaci lze využít co nejvěrnější modely fyzických komponent, čímž získáme simulaci, co nejvíce přiblíženou skutečnému světu. FluidSIM podporuje vzdělávání a učení a proto jsou všechny komponenty řádně zdokumentovány a graficky znázorněny včetně animací usnadňující pochopení jednotlivých zařízení. [42]

### 3.8 NX Mechatronics Concept Designer

NX je flexibilní a výkonná CAD/CAM/CAE platforma firmy Siemens, která byla do roku 2007 vlastněna firmou UGS Corporation. Poskytuje řešení v oblasti návrhu, simulace a výroby. Podporuje každou fázi vývoje produktu, od návrhu konceptu, přes technickou analýzu až po výrobu produktu. Tato platforma podporuje integraci s ohromným počtem různých nástrojů. Jedním z nich je i nástroj Mechatronics Concept Designer. [43; 44]



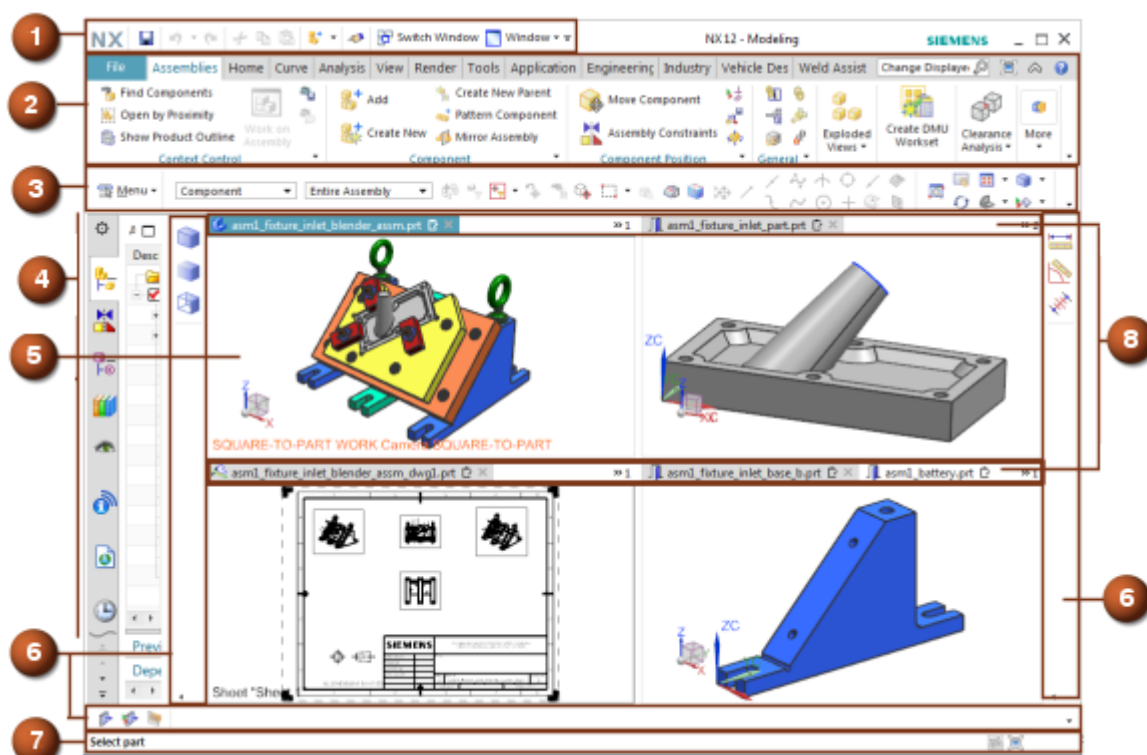
Mechatronics Concept Designer je nástrojem pro interaktivní návrh a simulaci komplexních pohybů elektromechanických systému. Podporuje vývoj stroje již od počáteční fáze, a to návrhem konceptu stroje včetně mechanických, elektrických, kapalinových a automatizačních prvků. MCD umožňuje paralelní mechatronický vývoj. To znamená, že různá oddělení spolupracují a postupují souběžně celým vývojem. Jedná se o propojení oddělení systémového inženýrství, konstrukce, elektroprojektce a programátorů. [44]

Za účelem virtuálního oživení lze do nástroje NX Mechatronics Concept Designer importovat konstrukční soubory vytvořené v jiných vývojových SW např. SolidWorks, Autodesk Inventor, CATIA aj. To nám umožňuje virtuálně oživit stroj, který byl zkonstruován i v jiném prostředí, než je software NX.

### 3.8.1 Uživatelské prostředí

NX MCD obsahuje rozsáhlé vývojové prostředí umožňující rychlý přístup k různým funkcím nástroje.

① *Panel rychlého přístupu.* Rychlý přístup k základním funkcím a příkazům jako je krok zpět nebo uložit. ② *Pás karet.* Uspořádané příkazy aplikací do karet a skupin ③ *Horní krajní panel.* Obsahuje příkazy nabídka, skupina výběru, skupina pohledů a skupina pomůcek. ④ *Panel zdrojů.* Obsahuje navigátor, palety a karty ⑤ *Grafické okno.* Dovoluje modelovat, vizualizovat a analyzovat modely. ⑥ *Levý, pravý a dolní krajní panel.* Zobrazuje uživatelské příkazy. ⑦ *Řádek nápovědy/Stavový řádek.* Vyzve uživatele k další akci a zobrazuje zprávy. ⑧ *Karty.* Zobrazuje název otevřeného souboru. [44]



Obrázek 3.10: Popis uživatelského prostředí NX [44]

### 3.8.2 Fyzikální vlastnosti










Pro požadované fungování modelu musíme potřebným dílům a soustavám přidat fyzikální vlastnosti. Modelům můžeme snadno přidat základní fyzikální chování, vytvořit kinematické řetězce pomocí různých typů kloubů a také omezit možnosti jejich pohybu. [44]

Tabulka 3.8: Fyzikální vlastnosti v NX MCD [44]

Vlastnost	Popis	Symbol
Nedeformovatelné těleso	Pohybová tělesa, na které mohou působit různé síly v systému, např. gravitace	
Kolidující těleso	Definice kolizí s jinými modely, která jsou také kolidujícím tělesem	
Zabránit kolizi	Zabránění kolizi mezi dvěma tělesy, tělesa mezi sebou proniknou	
Plocha přesunu	Vytvoření dopravníkového pásu z ploché roviny	
Zdroj objektu	Vytvoření několika instancí objektu, vhodný pro simulaci materiálového toku v systému	
Skladiště objektu	Odstranění objektů vytvořených zdrojem objektu	
Sledovač (Tracer)	Sledování dráhy bodu na nedeformovatelném tělese	
Kolidující materiál	Definice chování tělesa během kolize, může být přiřazen koliznímu tělesu nebo transportní rovině	
Změnit materiál	Definice chování kolize u dvou kolidujících těles	
Tlumič pružiny	Použití síly nebo točivého momentu na spoj	
Řízení síly/momentu	Připojení síly nebo krouticího momentu k ose spoje	
Měnič zobrazení	Změna vzhledu nedeformovatelného tělesa během simulace	

Tabulka 3.9: Typy spojů v NX MCD [44]







Typ spoje	Popis	Symbol
Otočení	Jeden stupeň volnosti – rotační	
Posuvný	Jeden stupeň volnosti – translační	
Válcový	Dva stupně volnosti – rotační, translační	
Šroubový	Jeden stupeň volnosti – rotační i translační zároveň	
Rovinný	Tři stupně volnosti – 1 rotační, 2 translační	
Kulový	Tři stupně volnosti – 3 rotační	
Pevný	Nula stupňů volnosti – pro spojení nedeformovatelného tělesa se zemí nebo dvou a více tuhých těles	
Úhlový s pružinou	Vytvoření pružinové síly při přemístění tuhého tělesa při úhlovém pohybu	

Lineární s pružinou	Vytvoření pružinové síly při přemístění tuhého tělesa při lineárním pohybu	
Křivka na křivce	Simulace těles valících se nebo klouzajících po křivce	
Bod na křivce	Sestava se pohybuje v závislosti na křivce, s kterou je propojen koncový bod sestavy	
Vazby dráhy	Nedeformovatelné těleso má omezený prostorový pohyb. Trajektorie je vytvořena řadou bodů, kterými bod nedeformovatelného tělesa prochází. K Simulaci robotických pohybů	
Úhlový s omezením	Zabránění pohybu těles se dostat nad danou limitní vzdálenost při úhlovém pohybu	
Lineární s omezením	Zabránění pohybu těles se dostat nad danou limitní vzdálenost při lineárním pohybu	
Podmínky zalomení	Definice maximální možné síly nebo momentu pro zlomení daného kloubu	
Virtuální osa	Vytvoření lineárních a úhlových spojů bez přiřazení geometrie. Vytvoření spojů a váček	
Řetězový	Propojení několika instancí pantových kloubů pro vytvoření řetězu	










### 3.8.3 Pohyb a senzory

Poté co je nastavena kinematika mechatronického systému je dalším krokem přidání ovládacích prvků pro řízení jednotlivých částí stroje. NX MCD umožňuje přidat do systému zobecněné akční členy. NX MCD obsahuje dva základní typy akčních členů, a to pro řízení rychlosti nebo polohy. Mezi pokročilé akční členy patří válce a ventily, jak hydraulické, tak i pneumatické. Po přidání akčního členu a senzorů je možné vytvořit senzor motorovou listinu již na začátku návrhu procesu. Přidání pohybových a senzorových vazeb je nezbytné pro budoucí napojení na signály z externích nástrojů. [44; 45]

Tabulka 3.10: Akční členy v NX MCD [44]

Akční člen	Popis	Symbol
Řízení polohy	Přesun kloubu v ose do definované pozice při definované rychlosti	
Řízení rychlosti	Nastavení rychlosti pohybu kloubu v ose na předem definovanou rychlost	
Hydraulický válec	Slouží pro přidání vlastností hydraulického válce, jako je plyn, píst a komory k posuvnému nebo válcovému kloubu	
Hydraulický ventil	Umožňuje přidat vlastnosti hydraulického ventilu k jednomu nebo více hydraulickým válcům, pro řízení průtoku	
Pneumatický válec	Slouží pro přidání vlastností pneumatického válce, jako je plyn, píst a komory k posuvnému nebo válcovému kloubu	
Pneumatický ventil	Umožňuje přidat vlastnosti pneumatického ventilu k jednomu nebo více pneumatickým válcům, pro řízení průtoku	

Tabulka 3.11: *Senzory v NX MCD* [44]

Senzor	Popis	Symbol
Akcelerometr	Snímač zrychlení pro získání úhlových nebo lineárních akcelerací nedeformovatelného tělesa	
Kolizní	Poskytuje zpětnou vazbu o interakci těles během simulace	
Vzdálenostní	Informuje o vzdálenosti nejbližšího kolidujícího tělesa	
Obecný	Vytvoří výstup pro jakýkoliv runtime parametr z fyzického objektu	
Inklinometr (Sklonoměr)	Snímač náklonu generuje orientaci nedeformovatelného tělesa v prostorových úhlech	
Koncový spínač	Vytvoří booleovský výstup při dosažení dané koncové polohy	
Snímač polohy	Poskytuje informaci o poloze akčního členu	
Relé	Vytvoří booleovský výstup s dvěma možnými pozicemi a hysterezí	
Snímač rychlosti	Poskytuje informaci o rychlosti akčního členu	

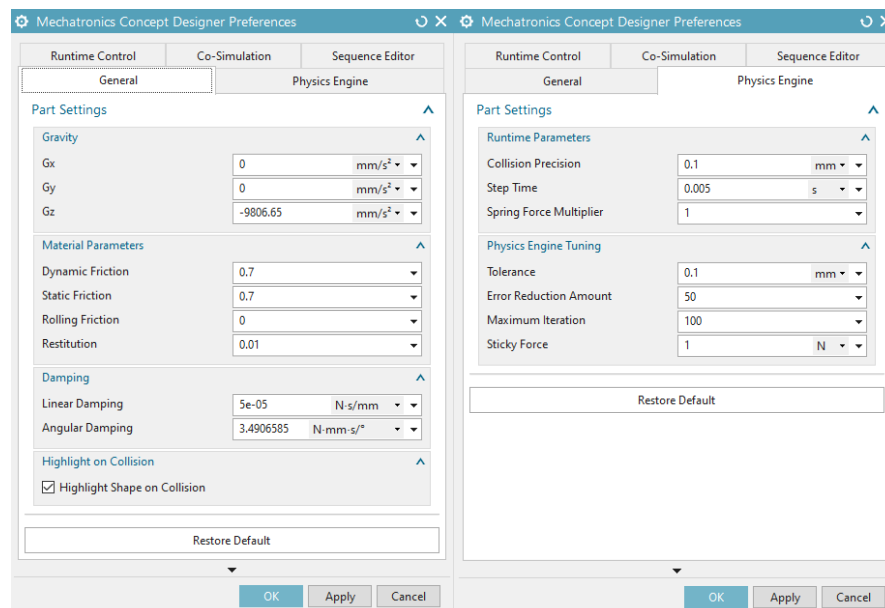
V nástroji lze také vytvářet složitější pohyby s váčkami a ozubenými koly. Lze zde simulovat koordinované pohyby několika os zároveň. S použitím funkce profil pohybu je jedna osa master a ostatní slave osy jsou s master osou pomocí pohybové křivky koordinovány. S ostatními funkcemi můžeme simulovat jednoduché převody mezi ozubenými koly, převod mezi ozubeným hřebenem a pastorkem nebo simulovat rotační pohyb motoru převádějící svůj pohyb na pásový dopravník. [44; 45]

### 3.8.4 Typy simulace

Po úspěšném přiřazení akčních členů a senzorů je možné vytvořit time-based nebo event-based simulaci. Time-based je simulace při které se akční členy pohybují v závislosti na čase. V daný časový okamžik např. v druhé vteřině simulace vyjede válec. Event-based simulace využívá k simulaci události. Jako událost lze označit např. změna stavu senzoru z log.0 na log.1., nebo změna externího signálu. Pro provedení pohybu je zapotřebí vytvořit nějaký řídicí element, k tomu poslouží příkaz *Operace*. Operace určuje, co se má stát za pohyb, může změnit podmínky nebo zastavit simulaci. Již zmíněné typy simulací umožňují vytvořit sekvenci operací a „rozhýbat“ strojní zařízení bez potřeby automatizačního modelu. Sekvence těchto operací lze exportovat a např. za pomoci vytvořeného generátoru automatizačního kódu danou sekvenci vygenerovat. [44]

### 3.8.5 Parametry simulace

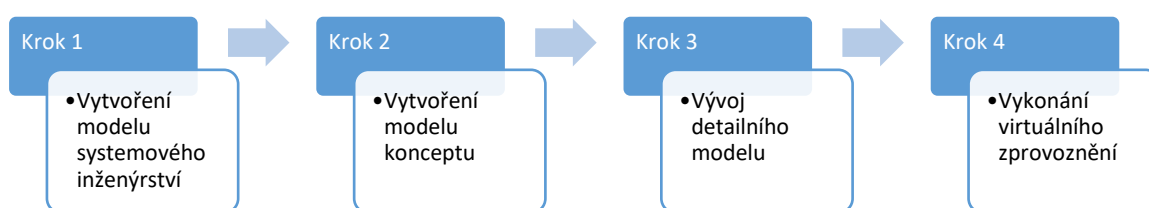
Ovládání simulace v NX je řešeno přes *Pás karet*. Lze zde spustit, ukončit, pozastavit, nastavit rychlost simulace nebo krokově vykonávat vytvořené operace. Parametry simulace lze upravit v okně *Mechatronics Concept Designer Preferences*. V základním nastavení může být upravena působící gravitační síla na všechna nedeformovatelná tělesa, různé druhy činitele tření nebo činitel restituace při kolizi dvou objektů a lineární a úhlové tlumení. V sekci *Physics Engine* lze upravit parametry jako je přesnost kolize, perioda výpočtů simulace, násobek pružné síly a parametry pro ladění enginu. [44]



Obrázek 3.11: Základní nastavení simulace a Physics Engine v prostředí NX

### 3.8.6 Workflow

Mechatronics Concept Designer udává přesný pracovní postup pro vytvoření modelu od jeho úplného návrhu za účelem virtuálního zprovoznění. Prvním krokem je začít s tvorbou modelu systémového inženýrství. Tento model je většinou tvořen v nástroji Teamcenter a do Mechatronics Concept Designer je poté importován. Nejprve je zde nutné specifikovat požadavky na stroj a vypsání cílů tohoto stroje. Následuje tvorba funkčního modelu, kde je vypsán seznam úkolů, které je nutno provést pro naplnění cílů. Poté je zapotřebí vytvořit logický model, který obsahuje seznam mechanismu potřebných pro dokončení požadovaných úkolů. Na závěr je potřeba vytvořit závislosti mezi požadavky, funkčním a logickým modelem a NX částí. [44]



Obrázek 3.12: NX Mechatronics Concept Designer workflow [44]

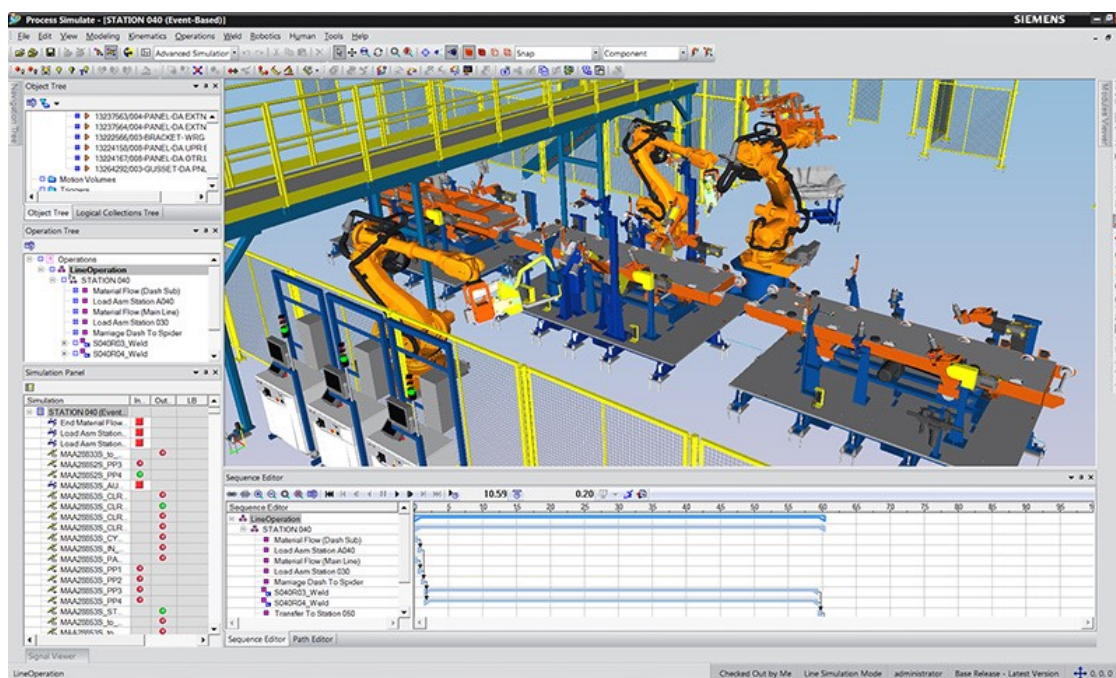
Při vytváření modelu konceptu jsou ze začátku importovány modely systémového inženýrství. Následně je vytvořen konstrukční model nebo dojde k jeho otevření. Na model jsou aplikovány základní fyzikální vlastnosti jako *Nedeformovatelné těleso* a *Kolizní těleso*. Jsou vytvořeny spoje tzn. úhlové a lineární pohyby sestav. Dojde k přidání akčních členů pro umožnění řízení pohybů. Následuje umístění senzorů do různých částí stroje. Pro zabránění nežádoucího chování je dobré nastavit nulové rychlosti u pohonů. V tuto chvíli je možné vytvořit simulaci za pomoci operací. Po simulaci můžeme exportovat profily motoru a senzor motorovou listinu. Pokud systém pracuje správně, je model konceptu dokončen. [44]



Vývoj detailního modelu vychází z již vytvořeného modelu konceptu. Při tvorbě začneme nahrazením koncepčních modelů za realistické modely sestav. Přidáme motory z databáze. Za pomoci runtime parametrů a výrazů dojde k vytvoření MCD signálů a následně budou použity jako vstupy a výstupy pro řízení simulace. Vytvořením Ganttova diagramu dojde ke spojení a vylepšení operací. Importem senzorů a akčních členů z ECAD synchronizujeme daný stroj s dokumentací. Na závěr je spuštěna simulace a v případě bez problémového chodu je detailní model hotov. [44]

Virtuální zprovoznění je posledním krokem, ve kterém je otevřen detailní model a jsou na něm aplikovány podrobnější charakteristiky pohybu. Následně je exportován list MCD signálů společně se sekvencemi operací. MCD signály jsou deaktivovány a signály jsou namapovány na externí SW. V posledním bodě je spuštěna kosimulace s externím nástrojem a virtuální zprovoznění stroje už zaleží na řídicí aplikaci. [44]

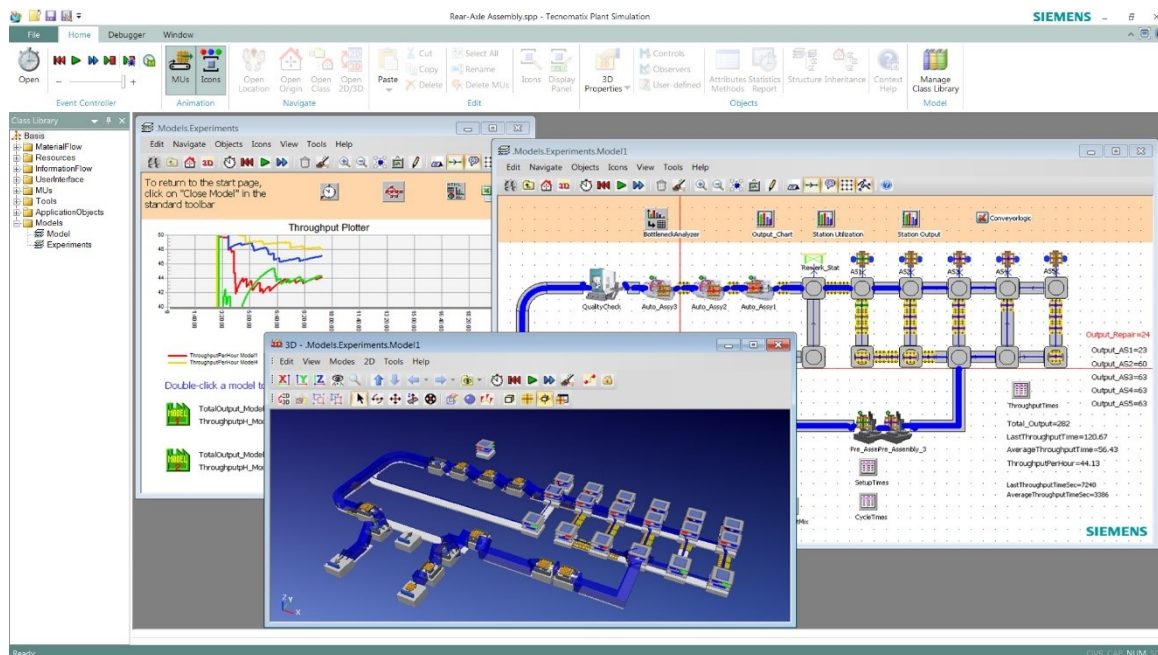
### 3.9 Tecnomatix Process Simulate



Obrázek 3.13: Vývojové prostředí Tecnomatix Process Simulate [46]

Nástroj Tecnomatix Process Simulate se zabývá primárně simulací robotických buněk a usnadňuje návrh a ověření výrobních procesů ve 3D prostředí. Umožňuje ověřit, specifikovat a validovat výrobní procesy včetně optimalizace doby cyklu a sekvencí. V prostředí lze simulovat celá sekvence sestavení produktu a také nástroje k identifikaci případných problémů nebo neefektivnosti. Každý aspekt výrobního procesu může být ověřen např. montáž, lidská práce, svařování, lepení a jiné robotické procesy. Velkým plusem je možnost emulace robotických kontrolérů a jejich teach pendantu. Je využito OLP (Offline Robot Programming) a programy různých výrobců robotů tak mohou být vytvářeny v prostředí Process Simulate. Logiku PLC programu jsme schopni ověřit připojením externího simulačního nástroje nebo fyzického řídicího systému. Výměna dat může probíhat pomocí komunikačního standardu OPC. [47; 48]

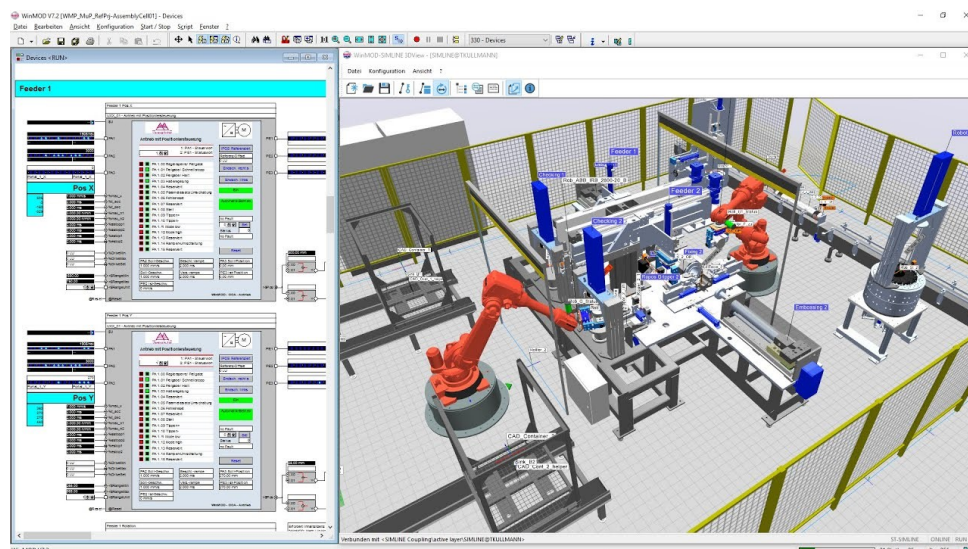
### 3.10 Tecnomatix Plant Simulation



Obrázek 3.14: Vývojové prostředí Tecnomatix Plant Simulation [49]

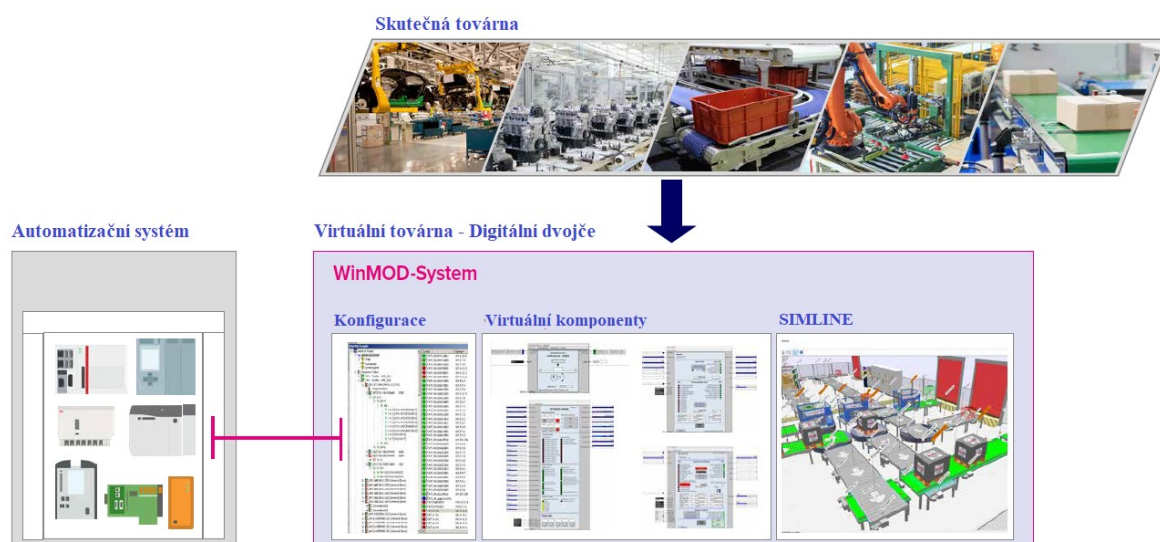
Tecnomatix Plant Simulation je simulační nástroj pro simulaci diskrétních událostí, který pomáhá vytvářet spolehlivé logistické a výrobní systémy. V prostředí Plant Simulation je možné prozkoumat vlastnosti systémů a rychle uskutečnit několik „co se stane, když ...“ scénářů. Mezi zajímavé výhody patří detekce a eliminace časově náročných a finančně nákladných problémů, zvýšení efektivity a přesnosti plánování a optimalizace logistiky a výkonu. Nástroj je možné použít na budoucí nebo již stávající továrnu. Simulační model využívá objektově orientovanou architekturu, která dovoluje vytvářet vlastní znovupoužitelné objekty a ukládat je do knihoven. Simulace může probíhat ve 2D nebo 3D prostředí. Otevřená architektura dovoluje snadný import a export souborů a nástrojů různého formátu jako je Excel, CAD soubory nebo XML. Je zde také možnost virtuálního zprovoznění propojením s řídicím systémem pomocí standardu OPC. Prostředí obsahuje různé nástroje pro analýzu. *Bottleneck analyzer* slouží pro rychlou identifikaci úzkých míst továrny. *Sankeyův diagram* je zde pro vizualizaci materiálového toku, což poslouží zejména pro nalezení přepravních a logistických problémů. Nemůže chybět ani *Ganttův diagram* pro vyobrazení využití zdrojů v čase. [50; 51]

### 3.11 WinMOD



Obrázek 3.15: Vývojové prostředí WinMOD [52]

Systémová platforma WinMOD vznikla již před 16 lety a po celou dobu své existence je úspěšně zaměřená na simulace různých druhů systémů. Momentálně se široce zaměřuje na svět průmyslové automatizace. WinMOD umožňuje vytvořit digitální dvojče skutečného zařízení a následně sledovat a simulovat chování tohoto systému v reálném čase. Po připojení k řídicímu systému lze virtuálně zprovoznit továrnu, linku nebo stroj. Jednou z největších předností nástroje je vysoký výkon při výpočtu behaviorálního modelu, kde je možné dosáhnout rychlosti až 1 ms. Výhodou je také možnost propojení s několika různými výrobci řídicích systémů. Pro simulaci chování různých zařízení WinMOD obsahuje knihovnu se skoro čtyřmi sty modely komponent. V určitých případech lze za nevýhodu považovat chybějící možnost časové synchronizace s jinými nástroji. Aplikace použití jsou např. různé druhy dopravníkových systémů, skladovací systémy, operace s průmyslovými roboty, paletování, třídění, balení a procesní automatizace. [15]



Obrázek 3.16: Koncept výrobní automatizace ve WinMOD [15]

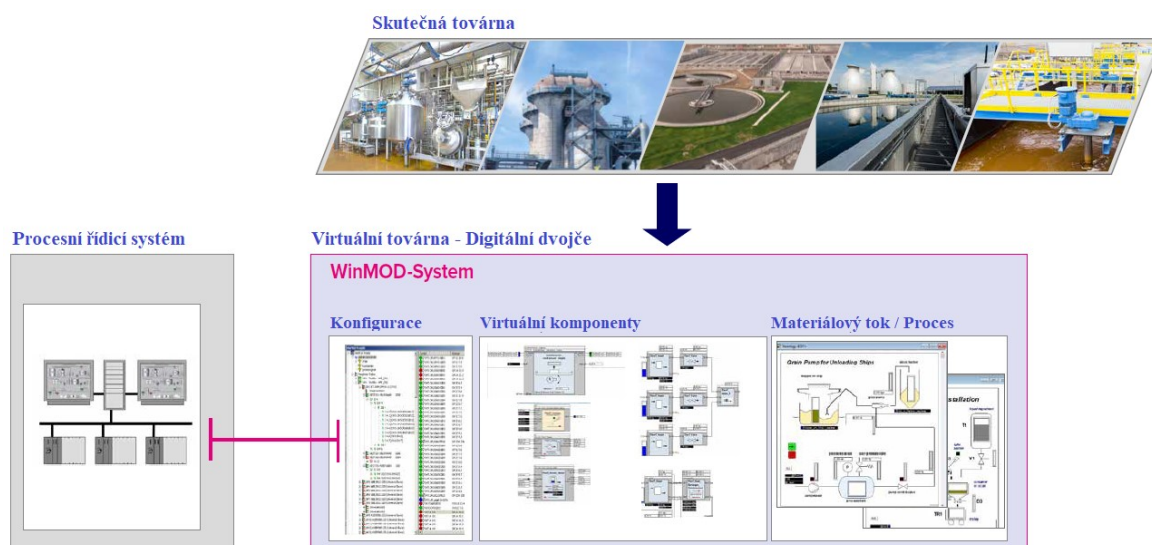


WinMOD systém se skládá ze 3 dílčích částí, konfigurace, virtuální komponenty a SIMLINE. V modelu virtuálního stroje je WinMOD schopen pokrýt celý prostor mechatronického systému, kde pokrývá elektrický a behaviorální model společně s fyzikálně-kinematickým modelem. [15]

V konfiguraci je řešeno připojení ke skutečnému nebo virtuálnímu řídicímu systému, je tedy podporována HiL a SiL simulační metoda. Výměna I/O signálů může probíhat s řadou automatizačních systémů různých výrobců jako je Siemens, Rockwell Automation, Phoenix Contact, Beckhoff, B&R, ABB, CodeSYS atd. Podporována je komunikace Profinet, Profibus, Interbus, Modbus, Ethernet/IP, MPI, OPC atd. U některých druhů komunikace může být potřebný dodatečný HW, např. komunikace pomocí Profinet a Profibus je možná pouze se příslušným simulačním boxem (WinMOD-PNIO Netbox, WM\_BX\_DP-xD). [15; 53]

Virtuální komponenty jsou modely určující chování daných skutečných zařízení. Jsou to zdigitalizována zařízení jako např. elektrické motory, frekvenční měniče, hydraulické válce, čtečky RFID kódů nebo čtečky čárových kódů. Můžou být označeny jako most mezi řídicím systémem a vyobrazeným virtuálním systémem. Jedná se v podstatě o elektrický a behaviorální model virtuálního stroje. [15]

SIMLINE je grafické vyobrazení fyzikálních vlastností a mechaniky zařízení a materiálového toku ve 3D prostředí, za pomoci modelu z knihovny nebo importovaných CAD souborů. WinMOD také umožňuje kosimulaci a je schopen pracovat s nástroji jiných výrobců jako je MATLAB Simulink, ABB RobotStudio, KUKA.OfficeLite, FANUC ROBOGUIDE, Microsoft Excel atd. V prostředí WinMOD je tedy možné ověřit i program a chování robotů výrobců ABB, KUKA nebo FANUC. [15]

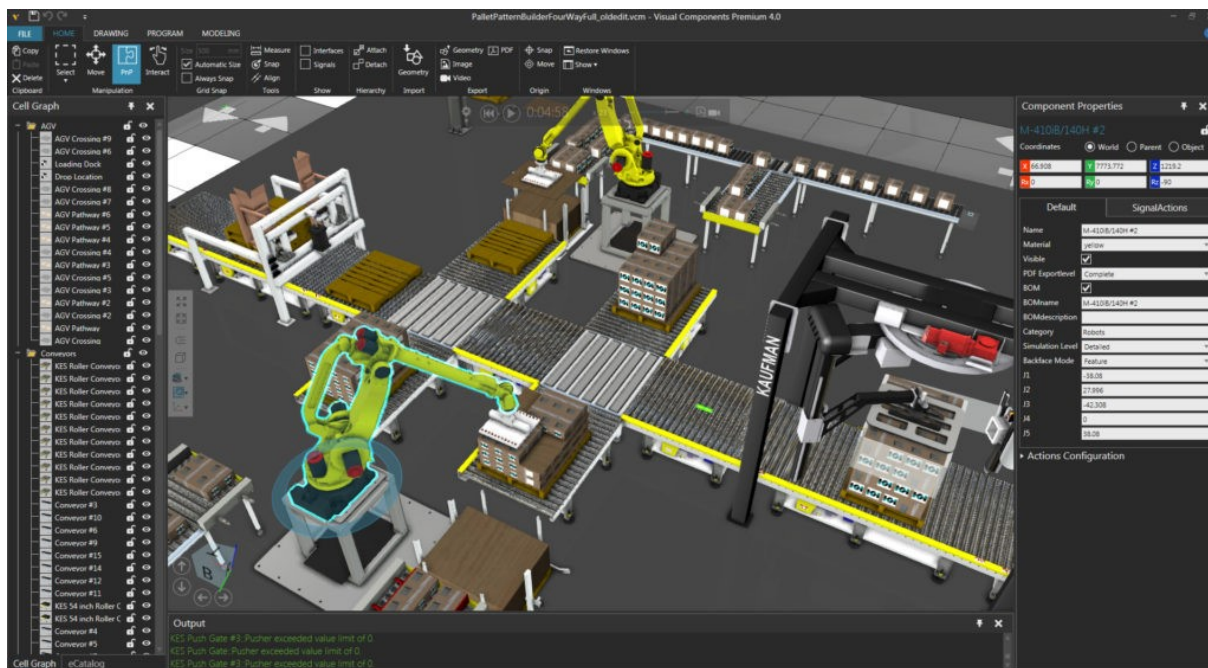


Obrázek 3.17: *Koncept procesní automatizace ve WinMOD* [15]

V případě použití nástroje WinMOD v procesní automatizaci, je v systému nahrazena část SIMLINE částí materiálový tok / proces. V procesní automatizaci totiž není kladen takový důraz na 3D svět a je zde důležité sledovat chování a vlastnosti např. různých kapalin, plynů nebo sypkého materiálu. Pro řízení může být použit procesní řídicí systém, který je napojen skrze konfiguraci WinMOD systému. [15]

### 3.12 Visual Components

Visual Components je 3D simulační software určený pro průmyslovou automatizaci. Je vyvíjen stejnojmennou firmou již od roku 1999. V současnosti je uznávaným globálním lídrem pro simulaci ve výrobním odvětví a jako důvěrný partner mnoha předních značek. Nástroj lze využít pro rychlý návrh a prezentaci výrobních řešení nebo jako platformu pro podporu off-line programování řídicích aplikací. Tento produkt je dostupný ve třech různých verzích podle velikosti obsahu (Essentials, Professional a Premium). [54]



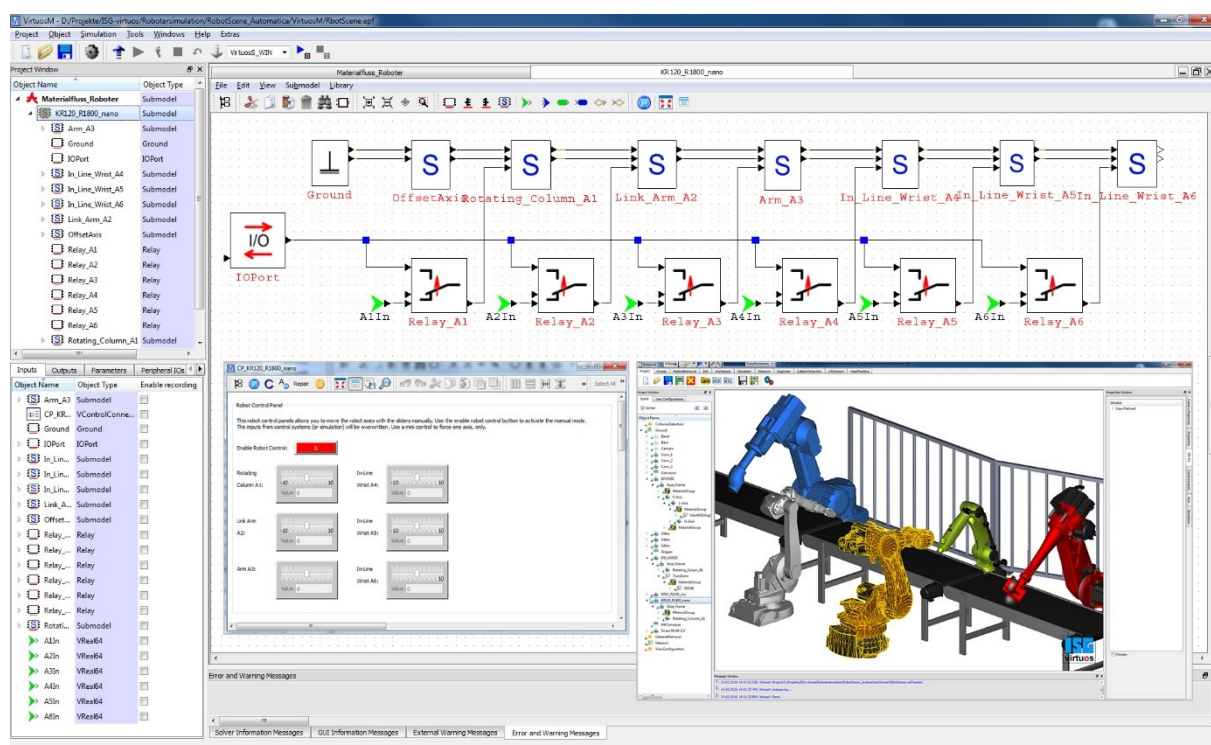
Obrázek 3.18: Vývojové prostředí Visual Components [55]

Mezi základní funkcionality patří intuitivní a rychlý návrh layoutu továrny, linky nebo robotické buňky. Pomáhá tomu eKatalog, což je knihovna komponent virtuálních modelů robotů, strojů a zařízení několika předních značek. Do 3D prostředí je možné importovat CAD soubory různých formátů. Modelování procesů je funkcionalita pro snadný a výkonný způsob pro vytvoření úloh v simulačním prostředí v již navrženém layoutu. Pro základní robotické simulace je zde balíček jednoduché robotiky umožňující joggování, ověření dosahů a kolizí a ujasnění logiky. Visual Components umožňuje připojení ke skutečnému nebo virtuálnímu řídicímu systému jako je PLC pomocí OPC UA nebo jiného podporovaného rozhraní. Prostedí také umožňuje sledovat simulaci pomocí prvků virtuální reality. [54; 55]

Vyšší verze produktu obsahují možnosti navíc, jako je úprava CAD modelů přímo v prostředí Visual Component, vizualizace lakovacího procesu a interaktivní virtuální realitu. Podporují off-line robotické programování OLP s nástroji pro urychlení a optimalizaci dráhy robotů a propojení s roboty značky Universal Robots a Stäubli za účelem ověření robotického programu. [54; 55]

### 3.13 ISG-virtuos

ISG-virtuos je simulační platforma vyvinuta firmou ISG, která byla poprvé představena v roce 2005 a zabývá se hlavně oblastí řízení a simulace strojních zařízení a továren. Platforma umožňuje ověřit a optimalizovat celý výrobní systém v reálném čase. Ověřena je logika řízení, pohyby robotů, materiálový tok atd. Simulační systém se skládá ze 3 nástrojů pro modelování (virtuos.M), výpočty (virtuos.S) a vizualizaci (virtuos.V). ISG-virtuos dovoluje využít simulačních technik MiL, SiL a HiL s deterministickým řízením v reálném čase. Doba odezvy 1 ms je zaručena v případě použití patřičného hardwaru. Je podporován import 3D CAD modelů v několika nejpoužívanějších formátech. Na počátku vývoje mohou být použity základní verze modelů, které jsou během vývoje postupně aktualizovány vydáním novějšími verzemi modelů. Nástroj podporuje kosimulaci s robotickými SW výrobci KUKA, FANUC, ABB aj. a nástroji pro simulaci řídicích systémů jako jsou PLC (SIEMENS, BECKHOFF) a CNC (SIEMENS). [56; 57; 58]



Obrázek 3.19: Vývojové prostředí ISG Virtuos [59]

### 3.14 Delmia V5

Simulační platforma vyvíjena společností Dassault Systèmes. Delmia V5 sdílí stejnou architekturu jako konstrukční nástroj CATIA. Platforma nabízí simulace hlavně v oblastech robotiky, ergonomie a plánování výrobních procesů. Největší předností je knihovna výrobních zařízení s více než 1000 průmyslovými roboty až 15 různých výrobců. Programy robotů mohou být vytvářeny přímo v prostředí Delmia, je zde využito OLP (Offline Robot Programming). Hotový program je pomocí překladače následně nahrán do fyzického kontroléru robota. Prostředí umožňuje simulovat operace robotů jako je obloukové i bodové svařování, lakování a manipulace s materiálem. Ověření,

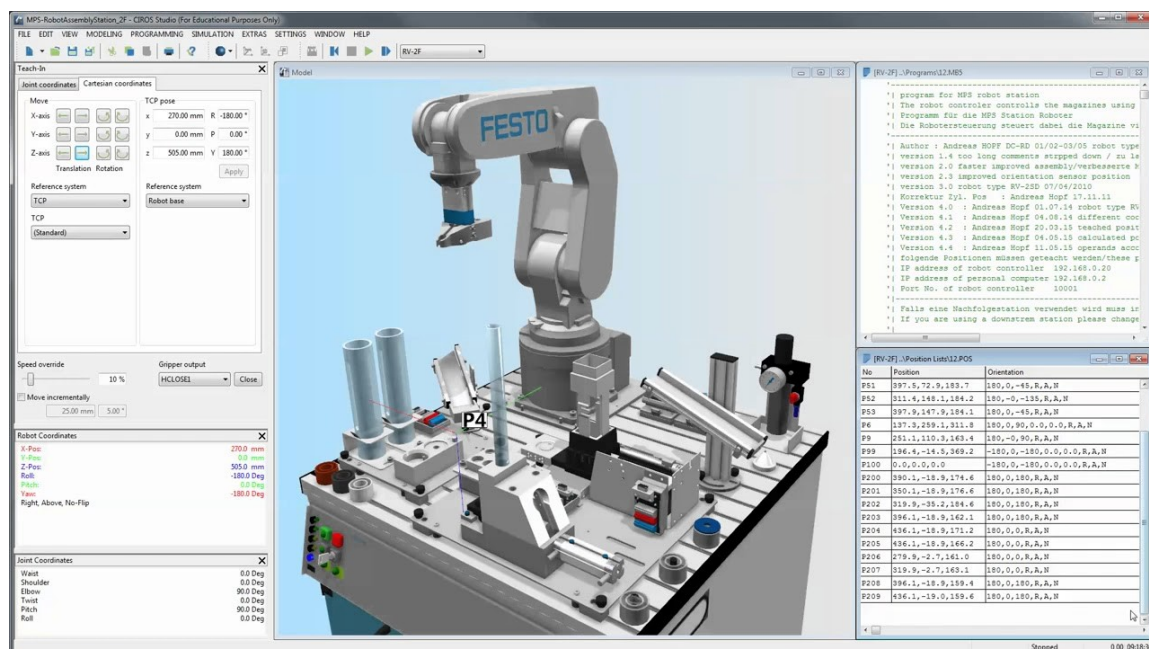


debuggování a testování PLC programu je také možné, jak s fyzickým, tak i se simulovaným PLC pomocí komunikačního standardu OPC. [60]



Obrázek 3.20: Vývojové prostředí DELMIA V5 [61]

### 3.15 CIROS Studio/Education



Obrázek 3.21: Vývojové prostředí CIROS Studio [63]

CIROS (Computer Integrated Robot Simulation) Studio je 3D simulační systém vhodný pro různé rozsahy použití. Vhodný je, jak ke vzdělávacím účelům (verze Education), tak pro realizaci

komplexních real-time simulací (verze Studio). SW umožňuje detailní návrh výrobních strojů a buněk, za účelem testování kritických situací, správné funkčnosti robotických a PLC programů a optimalizaci layoutu buňky. Všechny pohyby a manipulace s materiálem mohou být simulovány pro ověření možných kolizí a nalezení slabých míst vedoucí k optimalizaci doby taktu. Nástroj obsahuje knihovnu jejíž součástí jsou stroje, roboti, nástroje, dopravníkové pásy atd. Import CAD souborů je podporován pouze ve formátu STEP. CIROS Studio umožňuje programování robotů přímo ve svém prostředí v jazycích Melfa Basic IV, KRL, RAPID, V+ a IRL (Industrial Robot Language). Komunikace s externími řídicími systémy je dovolena pomocí rozhraní OPC klient/server, PARSIFAL a pro Mitsubishi roboty RCI Explorer (Robot Controller Interface). [62]

### 3.16 Factory I/O

Jedná se o 3D simulaci výrobní továrny za účelem vzdělávání v automatizaci. Ve virtuálním prostředí mohou studenti, nový zaměstnanci nebo školené osoby provádět testy a virtuálně oživit daný stroj nebo celou linku. Lze zde jednoduše a velice rychle vytvořit virtuální továrnu pomocí knihovny obsahující běžná průmyslová zařízení. Dostupných je také mnoho přichystaných aplikací s již rozvrženými zařízeními a stroji inspirovanými z průmyslu. Factory I/O je nejběžněji používáno společně s PLC řídicími systémy, nicméně může být použito i s mikrokontrolery nebo softPLC. Podporovány jsou PLC výrobců Allen-Bradley a Siemens. Komunikace probíhá přes klasické ethernet rozhraní, pomocí Modbus nebo OPC DA/UA. [64]



Obrázek 3.22: Vývojové prostředí Factory I/O [65]

### 3.17 Simulace technologických procesů

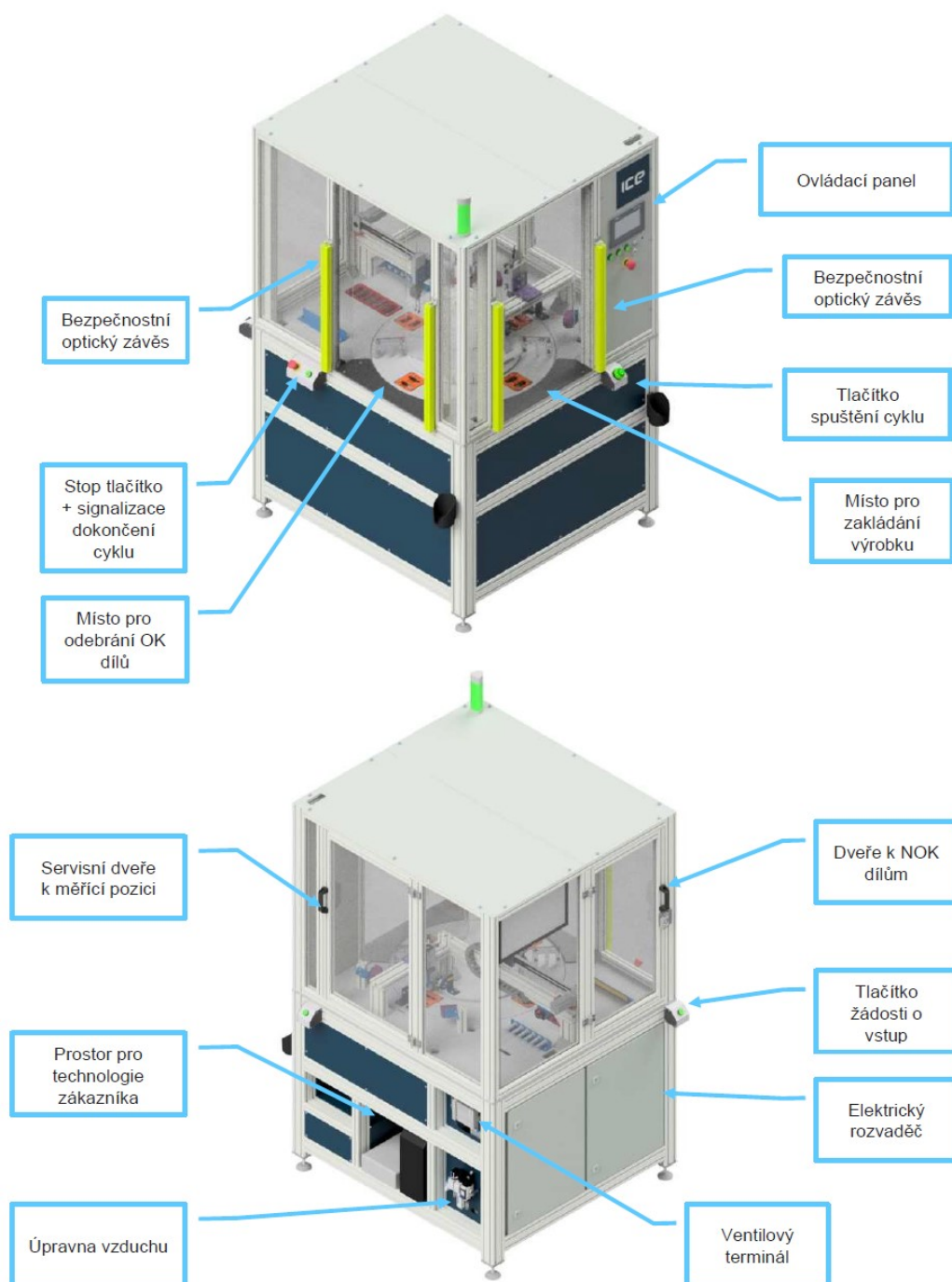
Pro simulaci a následnou optimalizaci technologických procesů slouží řada různých simulačních softwarů. Výpis několika zajímavých nástrojů je uveden v tabulce níže s přiřazením daného procesu k nástroji.

Tabulka 3.12: *Seznam nástrojů pro simulaci specifických technologických procesů*

	<b>Spojování</b>	<b>Svařování</b>	<b>Obrábění</b>	<b>Tváření</b>
ESI SYSWELD		<b>X</b>		
FORGE® NxT	<b>X</b>		<b>X</b>	<b>X</b>
LS-DYNA	<b>X</b>			<b>X</b>
NCSIMUL			<b>X</b>	
NX CAM			<b>X</b>	
Simufact Forming/Welding	<b>X</b>	<b>X</b>		<b>X</b>
SORPAS®	<b>X</b>	<b>X</b>		
VIRFAC®		<b>X</b>	<b>X</b>	

## 4. Popis pracoviště

Jednoúčelové strojní zařízení slouží k testování senzoru Ad-blue Bosch 5.3. Stroj je samostatný a nenavazuje na něj žádný další. Díly jsou zakládány na vstupu a hotové jsou ze stroje odebrány operátorem. Vlastní měření teploty termistoru senzoru je provedeno na PC zákazníka, které s PLC systémem komunikuje přes digitální vstupy a výstupy. Senzor slouží k měření teploty, hladiny a přítomnosti Ad-blue v nádržích na Ad-blue u dieselových aut.



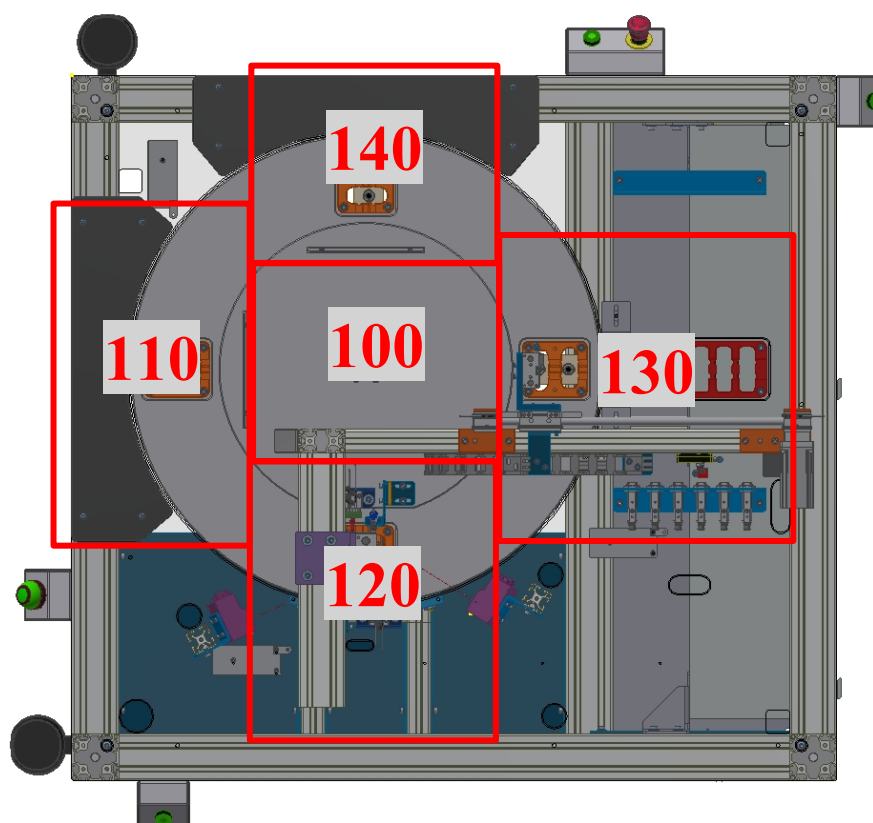
Obrázek 4.1: Model jednoúčelového stroje pro testování senzorů

Operátor založí jeden až dva senzory do lůžek, stiskne odbavovací tlačítko a stůl se otočí. Pneumatickým válcem dojde k zafixování senzorů, ze spodu vyjedou kontaktovací jednotky, které provedou elektrické spojení senzorů a proběhne jejich vyhodnocení na měřicím PC. Pokud měření dopadne OK značící válec provedou označení senzorů. Pokud je některý ze senzorů vyhodnocen jako NOK, dojde k jeho přemístění do zásobníku NOK dílů. V případě vyhodnocení senzoru jako OK je na výstupu vyjmut operátorem.

## 4.1 Layout

Stroj je konstruován jako kompaktní stanice. Z pohledu bezpečnosti se jedná o jednu zónu a 5 stanic neboli subzón.

- 100 – otočný stůl
- 110 – stanice zakládání dílů
- 120 – stanice měření
- 130 – stanice NOK dílů
- 140 – stanice vykládání OK dílů



Obrázek 4.2: *Layout a rozdělení stanic jednoúčelového stroje pro testování senzorů*

## 4.2 Hardware

Na skutečném stroji je použito několik různých zařízení.



## **PLC**

Stroj ovládá jediné PLC firmy Siemens – SIMATIC ET200SP 1510SP F-1 PN. PLC má 3 Profinet porty, 150 kB pracovní paměti a 750 kB pro data.

## **HMI**

Vizualizace běží na statickém panelu Siemens KTP 700 Basic. Jedná se o dotykové HMI s 8 dostupnými funkčními klávesami. Obsahuje Profinet rozhraní a dostupnou paměť 10 MB pro uživatelská data.

## **IO-Link zařízení**

Pro napojení čidel v poli jsou použity 3x slave IO-Linkové moduly Balluff BNI IOL-302-002-K006. Jako master IO-Link moduly jsou použity Siemens komunikační moduly CM 4xIO-Link.

## **Tlačítko pro ovládání sekvenčního režimu**

Stroj je vybaven tlačítkem Euchner ZSA1A5G05AC1917, které lze od stroje odpojit a slouží ke spuštění jednotlivých kroků automatické sekvence.

## **Čtečky 2D kódu**

Stroj je vybaven 2 čtečkami Keyence SR1000, které slouží k čtení kódů senzorů a jsou připojeny přímo do měřicího PC.

## **Pyrometry**

Stroj je vybaven 2 pyrometry Micro-Epsilon CT-SF15, které jsou připojeny přímo do měřicího PC a slouží k porovnání teploty termistoru.

## **Pohon Festo**

Manipulátor NOK senzorů využívá lineární osy od firmy Festo 1512320 CMMO-ST-C5-1-LKP. Komunikace s tímto modulem je po sběrnici IO-Link.

## **Ejektor Piab**

Pro držení NOK senzorů používá manipulátor 2 přísavky, vakuum se tvoří pomocí ejektoru Piab 3222067 X5A6-AN

## **Senzor vakua**

Zda je v přísavkách dosaženo správné vakuum pro zvednutí NOK senzoru kontroluje senzor vakua Piab 0110246 VS4015.

## **Úpravna stlačeného vzduchu**

Pro ovládání stlačeného vzduchu je použit hlavní ventil Festo 527709 MS4-EE-1/4-10V24P-S-AD7-1D.

### **Senzor tlaku vzduchu**

Dostatečný tlak vzduchu v systému hlídá senzor Festo 8001200 SPAU-P10R-MS4-F-L-PNLK-PNVBA-M12D. Hlídá také dosažení nulového tlaku vzduchu po vypnutí pneumatiky, dříve totiž nemohou být odemčeny dveře.

### **Ventilový ostrov**

Pro ovládání pneumatických válců je použit ventilový ostrov Festo 569926 34P-LK-SD-U10A-7K3L. Tento terminál je na sběrnici IO-Link.

### **Maják se sirénou**

Maják je napojený přes IO-Link a disponuje RGB barvami BNI IOL-801-000-Z037.

Osvětlení:

- Zelená – přerušovaně – předvolený automat
- Zelená – trvale – automat v chodu
- Bílá – trvale – manuální režim
- Modrá – trvale – servisní režim
- Červená – trvale – porucha

### **Tlačítko pro operátora**

Tlačítko pro operátora pro potvrzení dokončení operace. Banner K50PTAMGRY3Q, možnost využití 3 RGB barev.

Prosvětlení tlačítka:

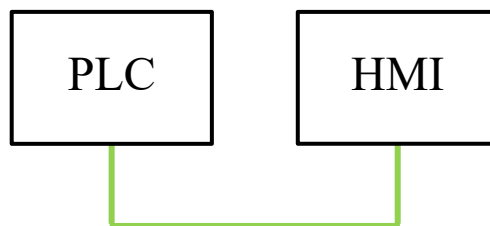
- Zelená – trvale – stroj v chodu
- Žlutá – trvale – prostor pro operátora
- Žlutá – přerušovaně – došel čas / čekání na potvrzení operace
- Červená – přerušovaně – porucha

### **Bezpečnost**

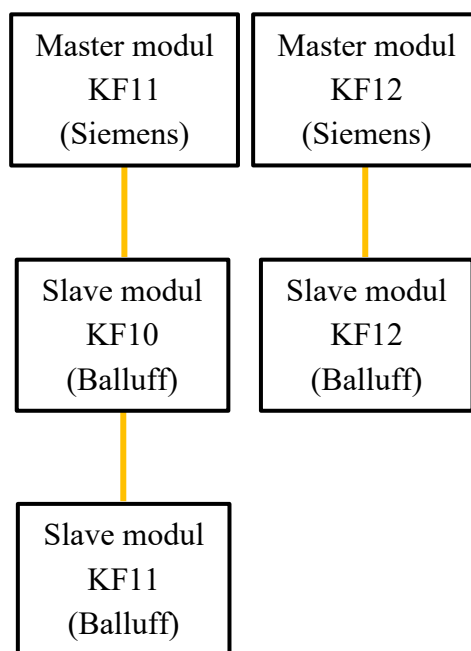
Stroj je vybaven dvěma tlačítky bezpečného zastavení. První je umístěno pod operátorským panelem a druhé na pozici vykládky OK senzorů. Zabezpečení a řízený vstup do pracovní stanice 120 hlídá Euchner dveřní zámek STM1N-222B024-M. Odemykání zámku je řízeno napětím. Pro jistění vstupu do zásobníku NOK senzorů slouží bezpečnostní senzor Balluff BID0008. Samotné dveře se uzamykají mechanickým zámkem. Pro zakládání a vykládání senzoru jsou na stroji umístěny dva páry bezpečnostních světelných závěsů Balluff BLG 4A-060-600-014-O01-SX. Všechny bezpečnostní prvky jsou připojeny přes HW IO.

## **4.3 Komunikace**

Komunikace mezi PLC a HMI panelem je provedena pomocí Profinet protokolu. Komunikace pomocí IO-Link je uskutečněna pomocí slučovačů pro vzdálené IO, řízení lineární osy Festo a majáku.



Obrázek 4.3: Topologie Profinet sítě



Obrázek 4.4: Topologie IO-Link sítě

PLC komunikuje s měřicím PC pomocí HW IO. Níže je uvedena tabulka výměny signálů.

Tabulka 4.1: Výměna signálu mezi PLC a měřicím PC (z pohledu PLC)

DO	DI
DUT1 in Position	DUT1 Test OK
DUT2 in Position	DUT2 Test OK
Test Start	SSI Test Done
DUT1 Mark OK	DUT1 Write Results OK
DUT2 Mark OK	DUT2 Write Results OK
Marking Done	Write Results Done
	Life Bit – 100ms

## 4.4 Ovládání

Na lince je k dispozici jeden statický panel KTP700 pro ovládání. Pro zapnutí stroje slouží 2 HW tlačítka. Po zapnutí hlavního vypínače začne startovat PLC i měřicí počítač. Pokud je PLC v režimu RUN a měřicí PC má aktivní *Life Bit*, rozbliká se kontrolka *Start* a kontrolka *Reset poruchy*. Na

vizualizaci zvolíme požadovaný režim, potvrdíme poruchu stiskem tlačítka *Reset poruchy*, čímž dojde k zajištění bezpečnosti a stiskneme tlačítko *Start*. Tím je stroj zapnutý a pracuje podle zvoleného režimu.

K dispozici budou tři hlavní režimy přepínatelné ve vizualizaci: ruční, automatický a sekvenční režim. Uživatel musí být pro změnu režimu autorizován, režimy budou voleny tlačítky operačního panelu.

### **Automatický režim**

Režim musí být vybrán na panelu a následně odstartován fyzickým odbavovacím tlačítkem. Je tedy rozdíl mezi zvolením režimu Automat a odstartování Automatu a tím zahájení produkce. Pokud je vybrán Automatický mód, všechny zařízení se nacházejí v automatickém režimu.

### **Ruční režim**

Režim musí být vybrán na panelu. Pokud je vybrán Ruční režim, zařízení se nacházejí v Ručním režimu. V Ručním režimu je možné pohybovat se všemi zařízeními z hlavního panelu za následujících podmínek:

- Pohybu nebrání blokující podmínky v SW, které chrání zařízení proti mechanickému poškození
- Bezpečnost stroje je v pořádku
- Pohyb více zařízení najednou je možný, pouze pokud patří k jedné stanici (nacházejí se na jedné obrazovce)

### **Servisní režim**

Režim musí být vybrán na panelu. Režim se chová podobně jako Ruční režim. Nejsou brány v potaz blokující podmínky v SW. Při nesprávném zacházení může dojít k mechanickému poškození stroje. V Servisním režimu je možné pohybovat se všemi zařízeními z hlavního panelu za následujících podmínek:

- Bezpečnost stroje je v pořádku
- Pohyb více zařízení najednou je možný, pouze pokud patří k jedné stanici (nacházejí se na jedné obrazovce)

### **Sekvenční režim**

Režim musí být vybrán na panelu. Režim se chová stejně jako automatický režim s tím rozdílem, že lze ovládat pouze jedno pracoviště, které je vybrané na obrazovce HMI a jednotlivé kroky automatického režimu se spouštějí po stisku tlačítka sekvenčního režimu (tlačítko na kabelu). Pro použití sekvenčního režimu musí být uživatel přihlášen jako „údržba“ -

### **Zvláštní režimy**

- **Vyprázdnění s testem**

Po navolení této funkce operátor na stanici 110 nekládá do stroje nové senzory, ale pouze odbavuje pracoviště. Tímto dojde k vyprázdnění stroje, přičemž senzory, které jsou již ve stroji jsou testovány. Pokud po zapnutí stroje jsou na stanici 120 díly a nemají dokončený cyklus – není Write Result OK, zkouška se opakuje od začátku.

- **Vyprázdnění bez testu**

Po navolení této funkce operátor na stanici 110 nekládá do stroje nové senzory, ale pouze odbavuje pracoviště. Tímto dojde k vyprázdnění stroje, přičemž senzory, které jsou již ve stroji nejsou testovány a manipulátor NOK dílů je odloží do zásobníku NOK dílů.

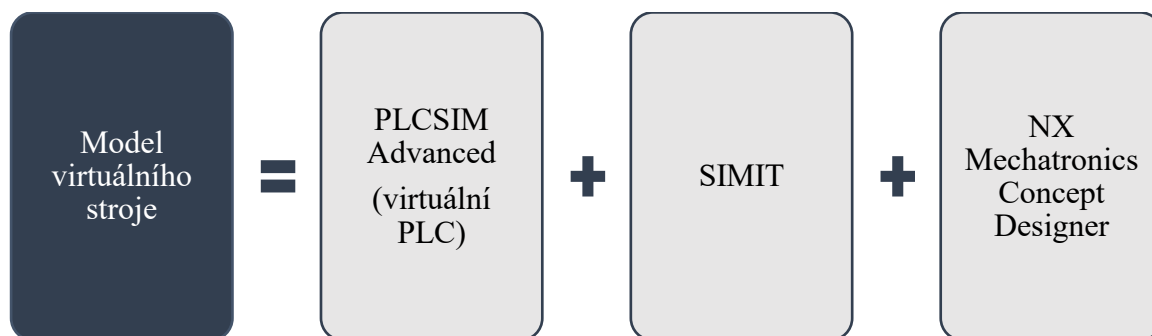
## 5. Postup řešení

Existuje několik způsobů virtuálního zprovoznění. Lišit se mohou v použitém softwaru, hardwaru, simulační technikou nebo typem komunikace. Základem simulace je popřemýšlet a uvědomit si jaké výsledky od ní očekáváme. V případě virtuálního zprovoznění jde to, jak moc detailně chceme stroj oživit, chceme oživit celou linku nebo pouze jedno zařízení, chceme otestovat PLC nebo robotické programy anebo obě zařízení a jejich vzájemný datový handshake, bude vytvořený dynamický virtuální model stroje využíván i po uvedení do provozu fyzického stroje? Na trhu jsou simulační nástroje, které mohou být využity pro zaškolení nebo otestování znalosti nových zaměstnanců, nástroje zaměřené spíše na robotické simulace s podporou OLP nebo komplexnější platformy umožňující „rozhybat“ statický CAD model celé výrobní linky do nejmenšího detailu. V této práci je požadováno virtuálně oživit jednoúčelový testovací stroj za účelem otestování a optimalizace programu, případně nalézt mechanické kolize.

### 5.1 Vytvoření modelu virtuálního stroje

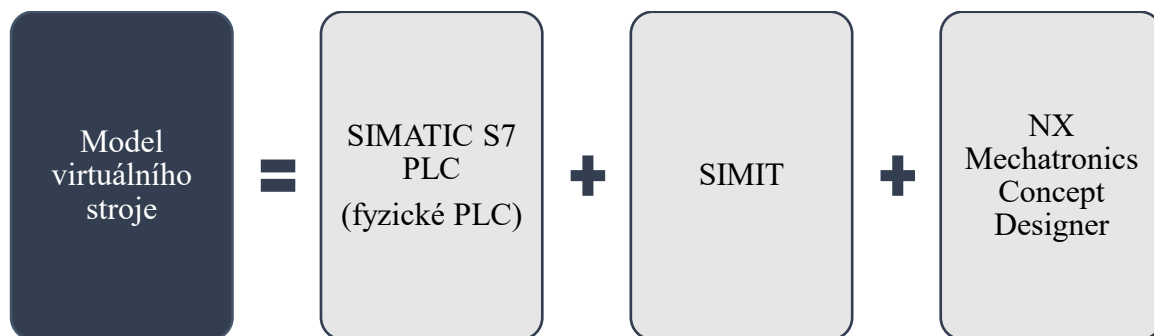
Pro virtuální zprovoznění testovacího pracoviště byly zvoleny nástroje, tak aby pokryly všechny oblasti modelu virtuálního stroje. Nejprve byl vybrán automatizační model. Protože pro řízení akčních členů bylo navrženo PLC SIMATIC od výrobce Siemens, je pro SiL simulaci ideální volbou nástroj PLCSIM Advanced. V případě vysokých požadavků na výkon PC je počítáno se simulací HiL, případně PiL. Zde by bylo vybráno fyzické Siemens PLC řady SIMATIC S7 stejného typu jako v dokumentaci.

Dalším krokem je volba fyzikálně-kinematického modelu, případně mechatronického modelu. Podle studia a prozkoumání trhu byly blíže přezkoumány 3 nástroje. Jedná se o nástroje WinMOD, Visual Components a NX Mechatronics Concept Designer. Nástroj WinMOD je působivý především kvůli širokému pokrytí všech úrovní simulace, kompatibilitě s různými výrobci PLC, podpoře velkého množství komunikačních rozhraní a možností pokrytí také behaviorálního a elektrického modelu virtuálního stroje. Nástroj Visual Components je na první pohled graficky vyspělejší, vyniká intuitivním a jednoduchým prostředím, podporuje celou řadu robotů a využívá především komunikaci pomocí OPC. Nástroj NX Mechatronics Concept Designer je vhodný pro virtuální zprovoznění jednoúčelových strojů, graficky působí slušně, simulační technologie je založena na herním enginu s využitím reálné fyziky a podporuje kosimulaci s nástroji PLCSIM Advanced a SIMIT. Vzhledem k lepší kompatibilitě s PLCSIM Advanced a dostupnosti licence na univerzitě byl jako simulační nástroj zvolen NX Mechatronics Concept Designer.



Obrázek 5.1: Zvolený model virtuálního stroje pro SiL simulaci

Pro pokrytí poslední části modelu virtuálního stroje byl zvolen nástroj SIMIT, který je podporován oběma vybranými nástroji. Siemens nabízí balíček SIMATIC Machine Simulator pro virtuální zprovoznění, jehož součástí jsou právě nástroje SIMIT a PLCSIM Advanced.

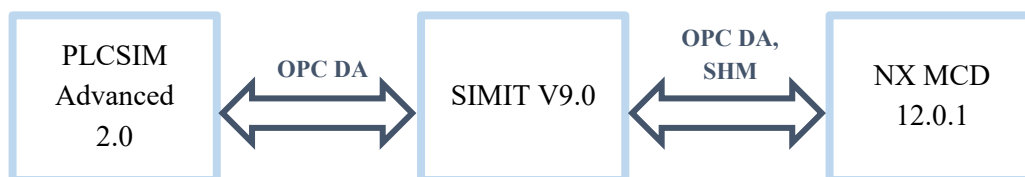


Obrázek 5.2: Zvolený model virtuálního stroje pro HiL, případně PiL simulaci

## 5.2 Výměna dat

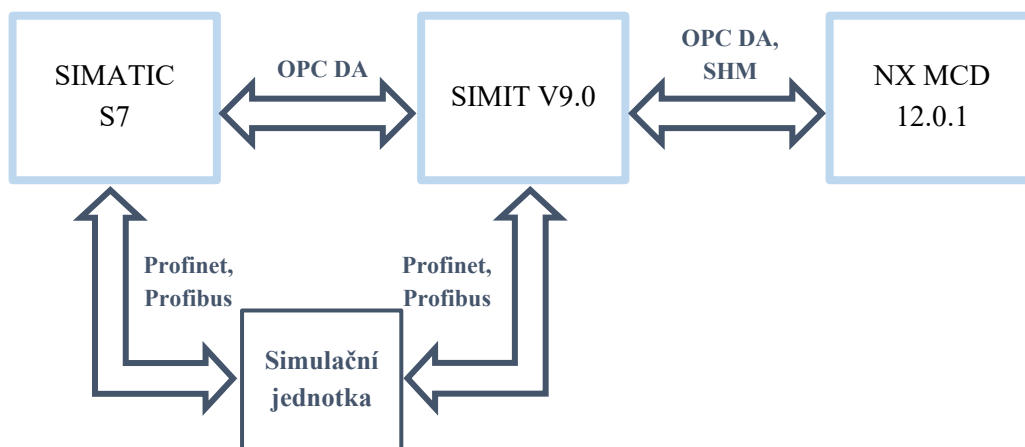
Pro komunikaci mezi jednotlivými SW nástroji je k dispozici několik způsobů. SIMIT je v tomto případě tzn. most a jeho komunikační dovednosti jsou zde klíčové. Jelikož novější verze SIMIT obsahuje více možností jsou zde popsány varianty komunikace s verzemi SIMIT V9 a V10.1.

### 5.2.1 Výměna při použití nástroje SIMIT V9.0



Obrázek 5.3: Výměna dat mezi modely pro SiL simulaci se SIMIT V9.0

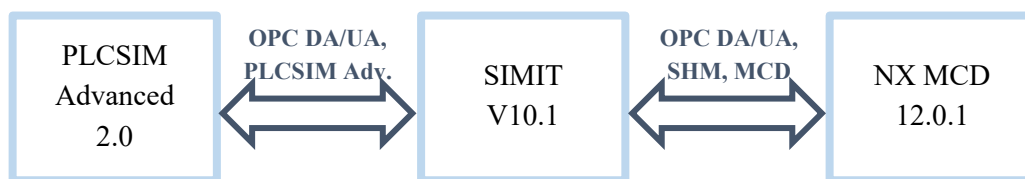
V případě SiL simulace se SW SIMIT V9 je možná výměna mezi všemi nástroji pomocí OPC DA standardu. Pro tento způsob je zapotřebí vytvořit OPC server, SW nástroje pak představují OPC klienty. SIMIT V9 a NX MCD mezi sebou také mohou vyměňovat data přes sdílenou paměť (SHM).



Obrázek 5.4: Výměna dat mezi modely pro HiL nebo PiL simulaci se SIMIT V9.0

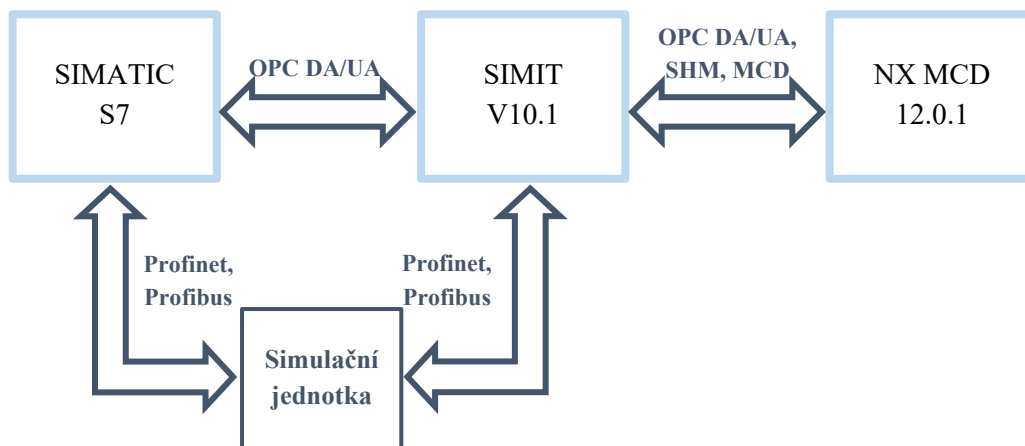
Pro simulaci HiL nebo PiL se SW SIMIT V9 je rozdíl v komunikaci oproti předchozí variantě pouze mezi automatizačním a behaviorálním modelem. Může být použito OPC DA, kde PLC komunikuje s nástrojem SIMIT pomocí OPC serveru. V tomto případě se jedná o PiL simulaci, jelikož nedochází k výměně I/O signálů, vstupní signály PLC není možné zapisovat, a tak musí být zapisováno do paměťových proměnných PLC. Druhou možností je Profinet nebo Profibus rozhraní. Jedná se o HiL simulaci. Zde je podmínkou vlastnit HW zařízení, sloužící jako simulační jednotka tzn. SIMIT UNIT (dříve označována jako SIMBA BOX). Při této komunikaci probíhá výměna dat mezi PLC a SIMIT stejně jako s reálným zařízením. Simulační jednotka je také schopná simulovat síťová zařízení. Podle typu jednotky je možné simulovat až 128 nebo 256 zařízení na Profinet síti nebo až 125 zařízení na Profibus síti.

## 5.2.2 Výměna při použití nástroje SIMIT V10.1



Obrázek 5.5: Výměna dat mezi modely pro SiL simulaci se SIMIT V10.1

S vyššími verzemi SW SIMIT je k dispozici více dostupných možností propojení. S příchodem SIMIT V9.1 byla vydána větší podpora kosimulace s nástrojem PLCSIM Advanced. S verzí V10 byla přidána podpora pro kosimulaci s nástrojem NX MCD a přidán nový pracovní mód synchronní sběrnice. To vede k pohodlnějšímu nastavení komunikace a možnosti použít synchronizovanou výměnu dat, kdy i při náročných simulacích získáme reálné chování systému. Od verze V9.1 je také podporován novější OPC UA standard.



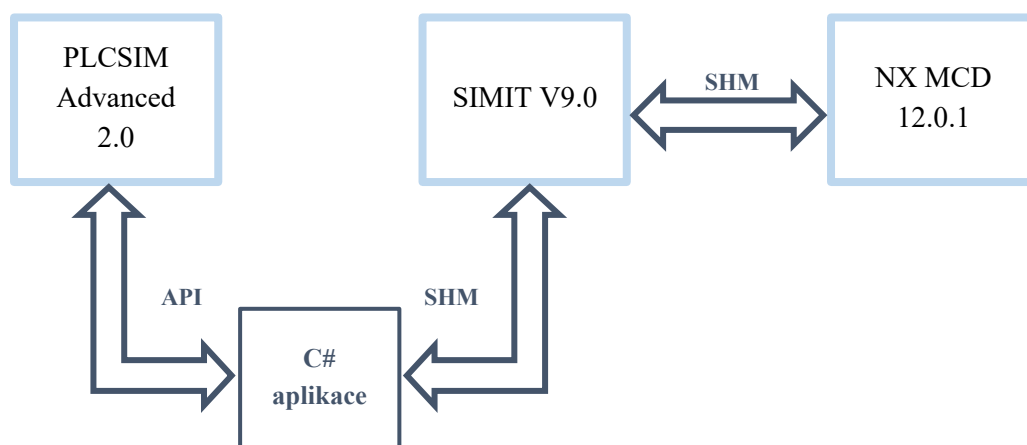
Obrázek 5.6: Výměna dat mezi modely pro HiL nebo PiL simulaci se SIMIT V10.1

## 5.2.3 Zvolený způsob

Vybrán byl způsob simulace SiL, pro kterou vystačí pouze PC. Jelikož na univerzitě byla k dispozici pouze starší verze SIMIT V9, způsoby dostupné u novější verze nemohly být použity. Výměna dat mezi NX MCD a SIMIT bude probíhat pomocí sdílené paměti (SHM). Podmínkou této komunikace je spuštěná simulace obou nástrojů na stejném PC. Prvotně bylo počítáno s výměnou dat mezi SIMIT a PLCSIM Advanced pomocí OPC DA protokolu. K založení OPC serveru byl použit program KepserverEx. Bohužel data ze simulátoru se nepovedlo dostat na server, a tak byla naprogramována aplikace, která přistupuje k proměnným simulátoru pomocí API. Data jsou aplikací



ukládána do sdílené paměti (SHM), do které také přistupuje nástroj SIMIT. Aplikace taktéž data zapisuje ze sdílené paměti do instance PLCSIM Advanced. Komunikace je tedy obousměrná. Konzolová aplikace byla vytvořena v IDE Visual Studio v jazyce C#.



Obrázek 5.7: Zvolený způsob výměny dat

### 5.3 Plánované kroky

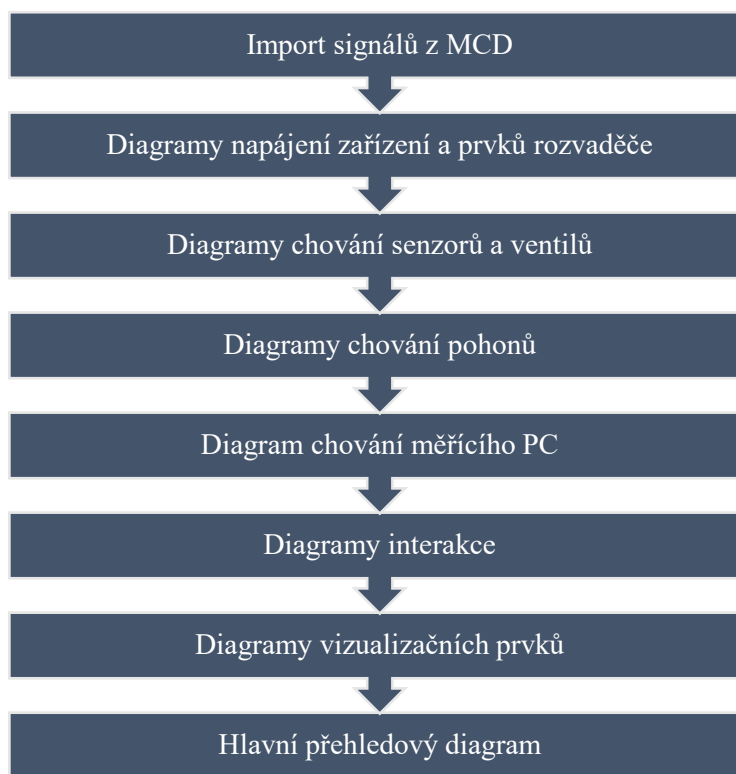
Pro zvolené řešení byl sepsán postup a ten byl rozdělen do 4 částí. První část se týká tvorby fyzikálně kinematického modelu v nástroji NX MCD. Druhá část se zabývá tvorbou behaviorálních diagramů v prostředí SIMIT. Třetí část se zabývá návrhem a tvorbou řídicí aplikace v prostředí TIA Portal. Poslední, čtvrtá část se zabývá nastavení výměny dat mezi modely, vytvořením konzolové aplikace a konečným virtuálním zprovozněním stroje.



Obrázek 5.8: Kroky pro vytvoření modelu v NX MCD

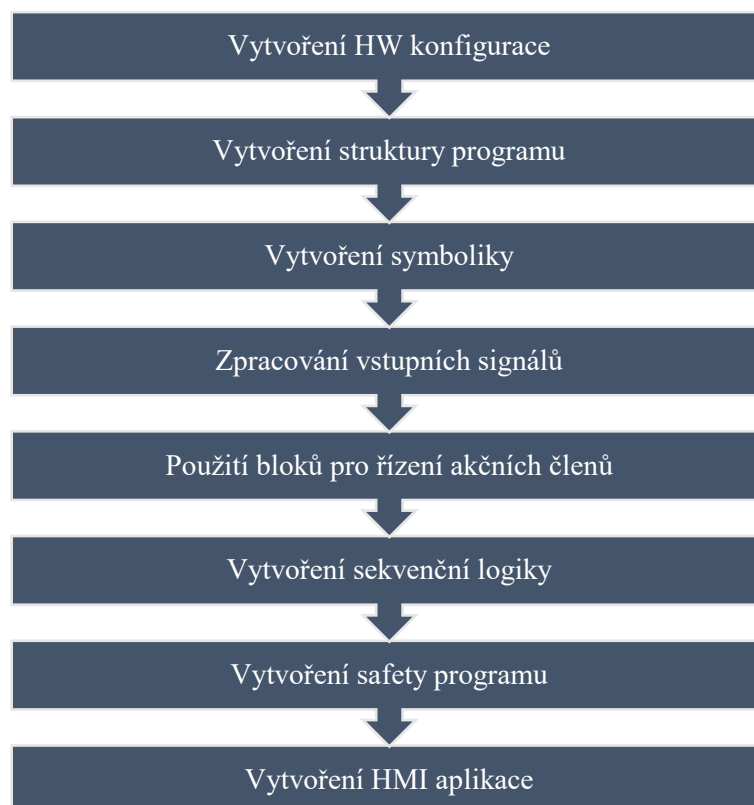
Po importování CAD modelu budou přiřazeny fyzikální vlastnosti jako je nedeformovatelné a kolizní těleso, a určeny typy spojů. Podle elektro dokumentace budou přidány všechny senzory závislé na tělesech a určeny pohony řízené polohou nebo rychlostí. V případě převodů jsou zde vytvořeny možné vazby. Bude nastaven zdroj a zánik objektů pro materiál a jeho přichycení vakuem manipulátoru.

Pro správnost fyzikálního a kinematického modelu bude vytvořena time-based simulace za pomoci sekvenčních diagramů. Následně proběhne vytvoření a připojení signálu na potřebné proměnné včetně exportu těchto signálů pro následný import do nástroje SIMIT.



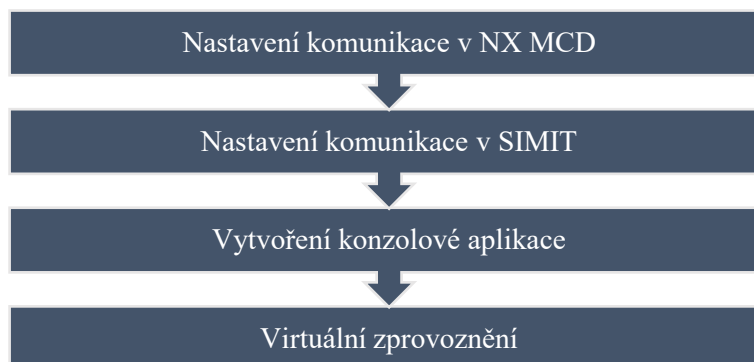
Obrázek 5.9: Kroky pro vytvoření modelu v SIMIT

V nástroji SIMIT bude nejprve proveden import signálů do vytvořeného SHM spojení. Jako první diagram bude vytvořen diagram pro prvky rozvaděče a vedení elektro a pneumatických rozvodů. Diagram chování senzorů bude vytvořen pro všechny senzory v elektro-dokumentaci. Senzory z NX MCD budou z větší části zrcadleny. Válcům bude vytvořen diagram chování podle pozice ventilu. Pro CMMO jednotku a lineární motor bude vytvořeno chování podle dokumentace výrobce k zařízení, stejně tak pro motor karuselu. Pro simulaci měřicího PC bude vytvořeno jeho chování a reakce na výstupy PLC. Diagram chování operátora bude obsahovat všechny možnosti interakce mezi strojem a člověkem např. stisk klasických nebo nouzových tlačítek, protnutí bezpečnostní závory, vkládání materiálu nebo otevření dveří. Diagram vizualizačních prvků bude obsahovat vizualizaci semaforu, indikátoru tlačítek a zpracování signálu pro zobrazení v NX MCD. Na závěr bude vytvořen hlavní přehledový diagram s přehledem hlavních prvků a pro snadnou interakci se systémem.



Obrázek 5.10: Kroky pro vytvoření řídicí aplikace v TIA Portal

Při vývoji řídicí aplikace bude nejprve vytvořena HW konfigurace podle elektro-dokumentace. Řídicí aplikace bude napsána ve firemním standardu MICE. Nejprve bude vytvořena základní struktura zóna, stanice a poté proběhne vytvoření datových bloků a merkerů. Dojde ke zpracování vstupních signálů a použití bloků pro řízení akčních členů. Pro každou stanici bude vytvořen sekvenční funkční blok zajišťující krokové provádění operací. Následně je vytvořen bezpečnostní program a na závěr vytvořena HMI aplikace.

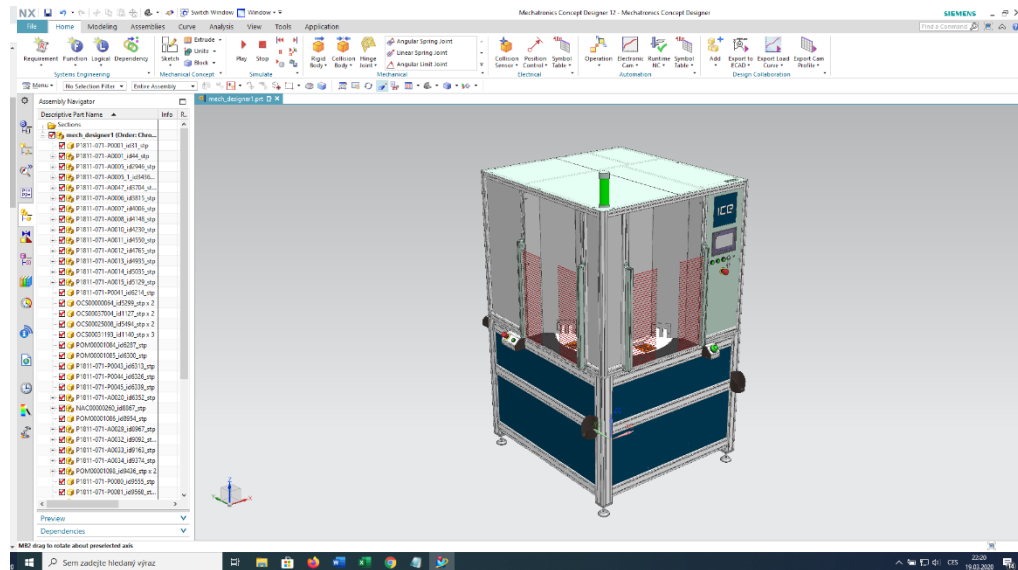


Obrázek 5.11: Kroky pro virtuální zprovoznění

Pro virtuální zprovoznění stroje bude nastavená komunikace v NX MCD a v SIMIT. Pro komunikaci mezi SIMIT a PLCSIM Advanced bude vytvořena konzolová aplikace. Na závěr proběhne virtuální oživení stroje a dojde k otestování a optimalizaci vytvořeného programu.

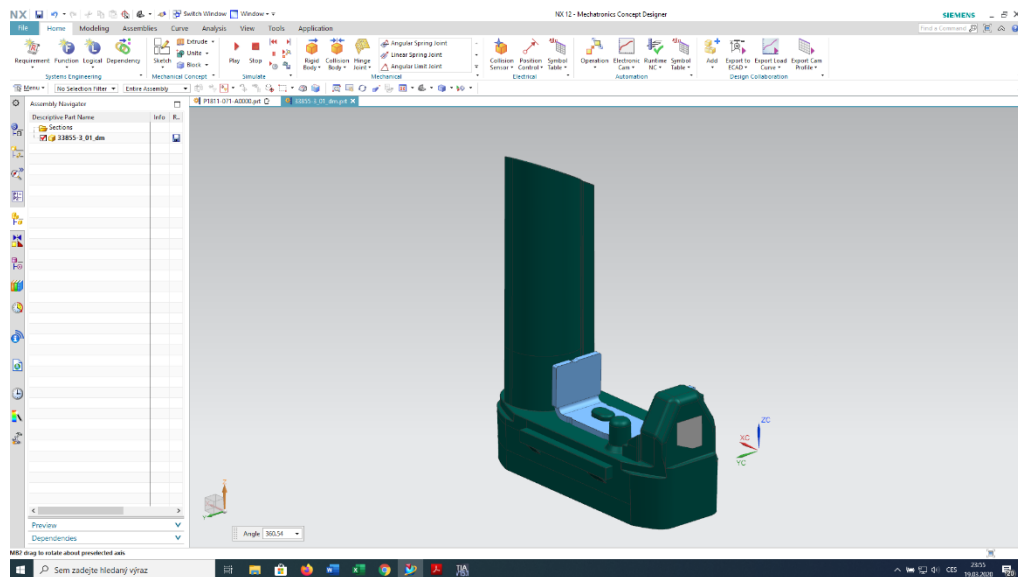
## 6. Realizace fyzikálně-kinematického modelu

Pro realizaci fyzikálně-kinematického modelu je potřeba vlastnit hotový 3D model stroje. Ten lze vytvořit v prostředí NX nebo v jiném konstrukčním nástroji a poté jej do NX nainportovat. CAD model byl vytvořen týmem konstrukce v prostředí Autodesk a poskytnut ve formátu STEP (.stp). V tomto případě byl proveden import modelu. Protože se nejedná o nativní formát prostředí NX, je k této operaci potřeba translator pro daný formát. Translatory různých CAD formátů jsou součástí instalačního balíčku společně s NX.



Obrázek 6.1: Importovaný CAD model stroje do prostředí NX

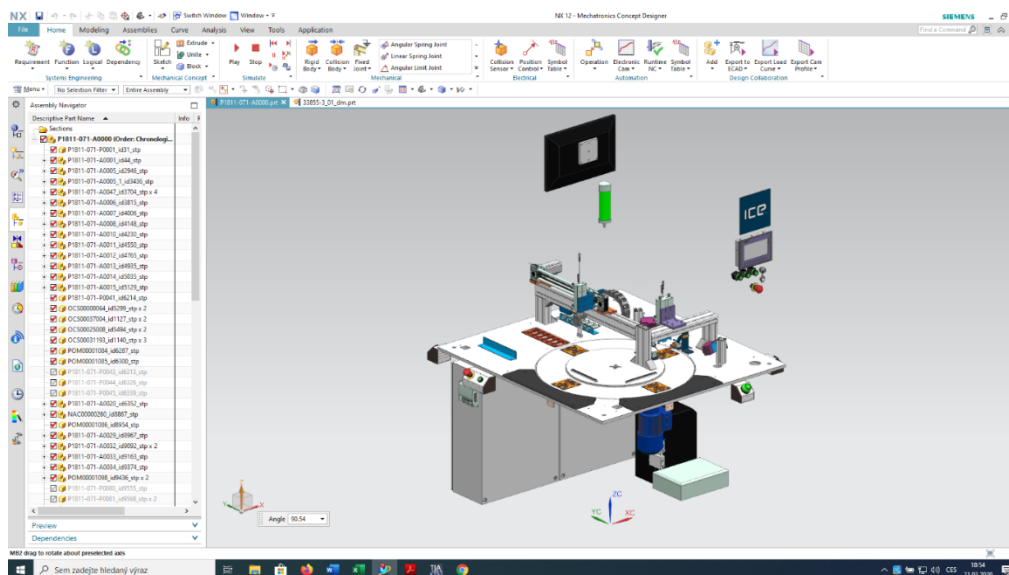
Týmem konstrukce byl také poskytnut 3D model senzoru ve formátu STEP. Jeho import proběhl stejným postupem.



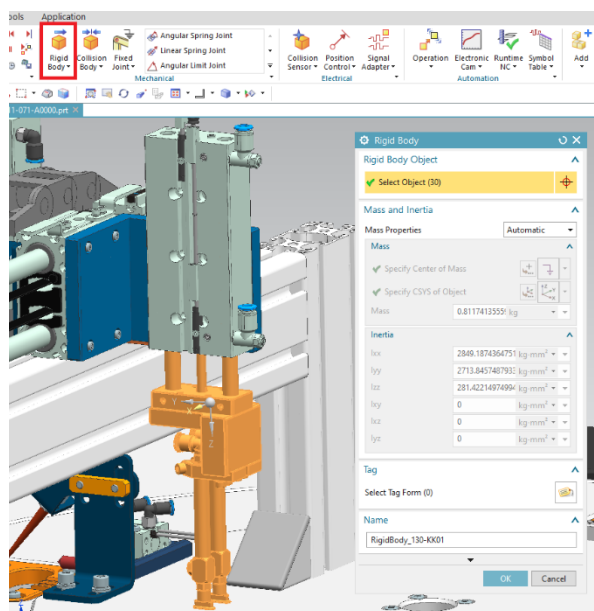
Obrázek 6.2: Importovaný CAD model senzoru do prostředí NX

## 6.1 Fyzikální vlastnosti

Před přidáním fyzikálních vlastností na dané části stroje, je vhodné skrýt statické části stroje, které jsou pro simulaci irelevantní. Patří zde různé kryty, plechy, profily, šrouby, matky aj. Model je poté přehlednější a fyzikální vlastnosti jsou přidávány jen tam kde to je potřeba, což také ušetří náročnosti simulace. Skrýt dané díly je možné přes *Panel zdrojů* v části *Navigátor sestav* pomocí checkboxu u dané sekce.



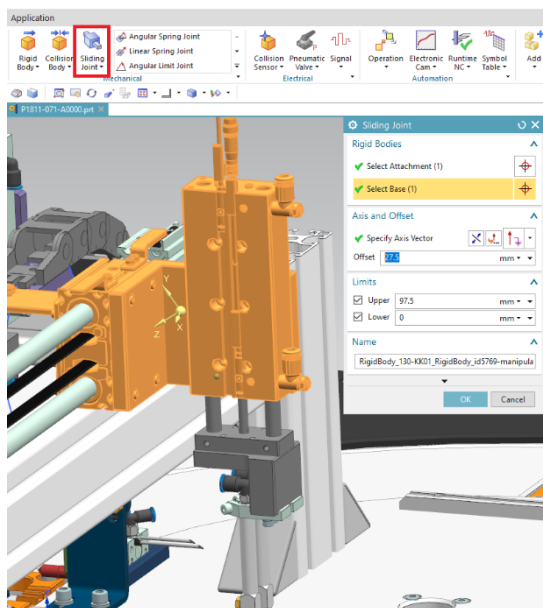
Obrázek 6.3: Odhalený model stroje v prostředí NX



Obrázek 6.4: Přidání nedeformovatelného tělesa v prostředí NX

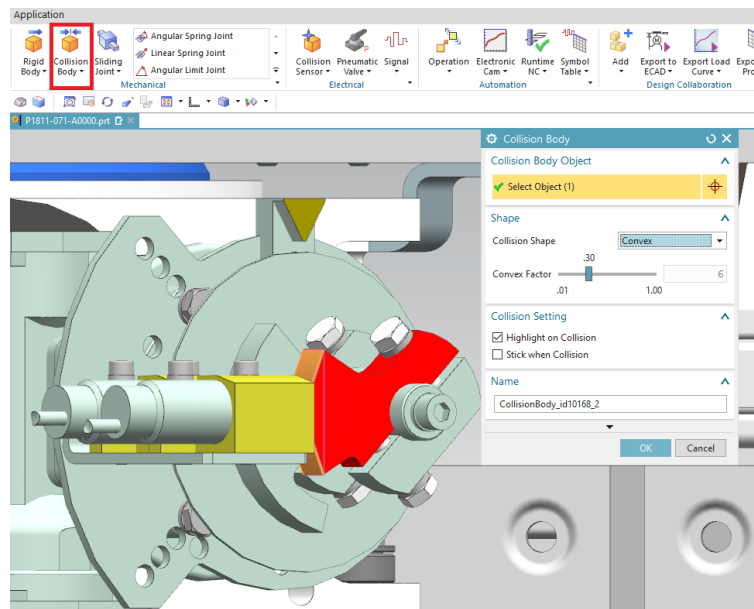
Pro přidání fyzikálních vlastností je potřeba v *Páso karet* vybrat *Nedeformovatelné těleso* a přes *Navigátor sestav* nebo *Grafické okno* nalézt a označit díly. Přes dialogové okno lze zadat hmotnost, těžiště a moment setrvačnosti vybraného tělesa. Parametry mohou být také vypočítány automaticky.

Jako nedeformovatelné těleso je nutné vytvořit všechny pohyblivé části modelu (válec, manipulátor, karusel, materiál). Na tyto díly bude působit gravitační síla.



Obrázek 6.5: Přidání posuvného spoje v prostředí NX

Pro vytvoření spoje, je potřeba označit vytvořené nedeformovatelné těleso a v *Páse karet* vybrat potřebný spoj, v daném případě se jedná o *Posuvný spoj*. V dialogovém okně je zvolena osa a případné omezení pohybu do obou směrů. Je možné zadat také offset, což značí výchozí pozici spoje při spuštění simulace. Základna je vybrána pouze pokud je vytvářen kinematický řetězec. V tomto případě je základnou osa X manipulátoru a připevněním je těleso pro pohyb v ose Z.

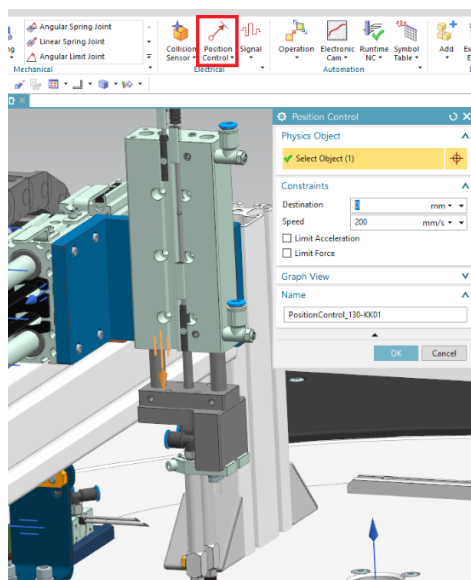


Obrázek 6.6: Přidání kolizního tělesa v prostředí NX

*Kolizní těleso* může být vytvořeno na celém nebo jen části povrchu dílu. Přes dialogové okno je možné zvolit z několika kolizních tvarů jako je kvádr, sférický, válec, kapsle, konvenční nebo mesh.

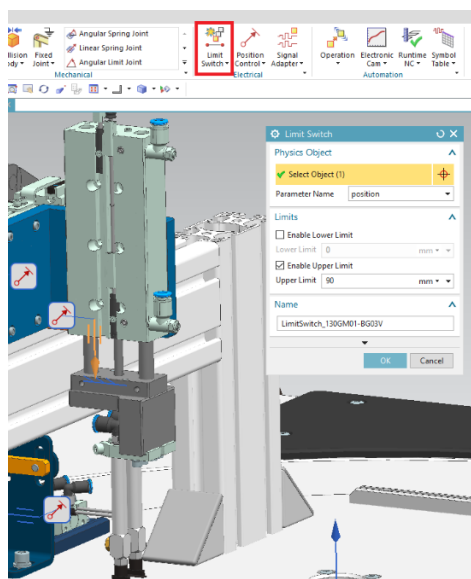
U komplexnějších tvarů je možné zvolit jejich faktor. Čím vyšší tento faktor bude, tím více bodů bude použito pro vykreslení kolizního tvaru. S rostoucím počtem bodů roste nejen přesnost, ale také náročnost simulace. Kolizní tělesa navzájem zabráňují průchodu objektu nebo mohou aktivovat kolizní senzor. V tomto případě bylo kolizní těleso vytvořeno na povrchu dílu upevněného na hřídeli motoru.

## 6.2 Pohyb a senzory



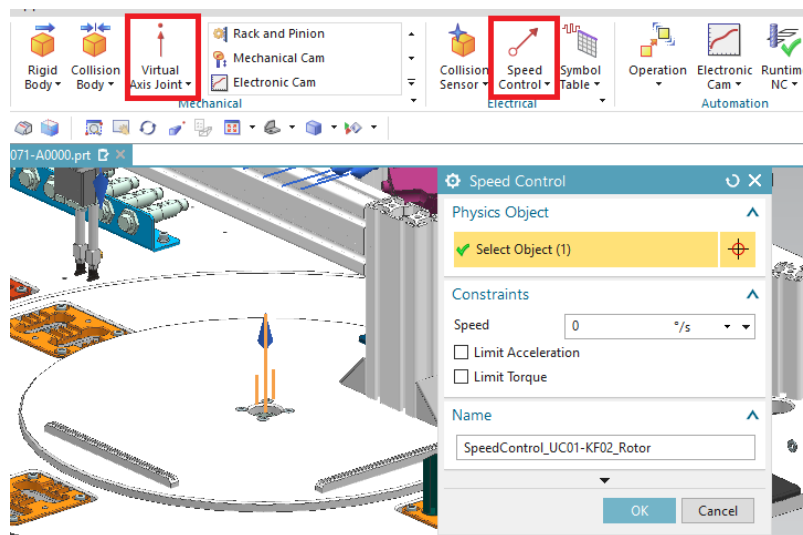
Obrázek 6.7: Přidání prvku řízení polohy v prostředí NX

Pro možnost řízení akčního členu je potřeba zvolit *Řízení polohy* nebo *Řízení rychlosti*, které se nachází v *Pásu karet*. V případě řízení polohy dojde k vyobrazení modrých šipek značící vytvořený spoj. Po označení spoje je zobrazeno dialogové okno, kde je možné zadat mezní hodnoty zrychlení nebo síly a zadat konstantní hodnotu rychlosti. Podobným způsobem lze přidat prvek *Koncový spínač* pro obě koncové polohy válce. V dialogovém okně je poté možné zadat horní nebo spodní mez spínače.



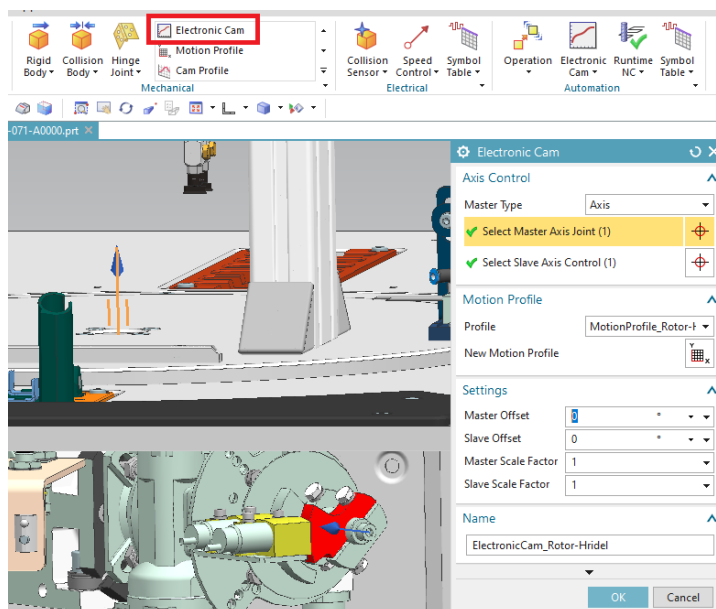
Obrázek 6.8: Přidání koncového spínače polohy v prostředí NX

Pro řízení rychlosti karuselu byla vytvořena *Virtuální osa*, představující rotor motoru. Změnou polohy rotoru je měněna poloha hřídele podle daného převodového poměru. Natočení karuselu je dáno polohou váčky podle určité charakteristiky. Tato charakteristika byla vytvořena pomocí prvku *Elektronická váčka*. Na rotor byl použit prvek řízení rychlosti a na hřídel a karusel prvek řízení polohy.



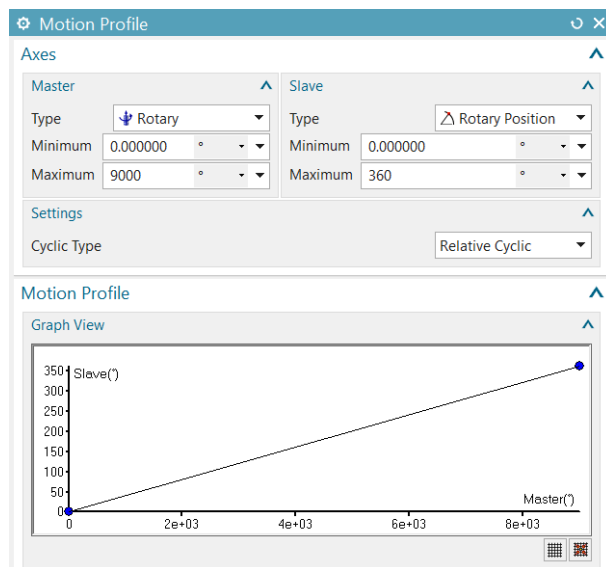
Obrázek 6.9: Přidání prvku virtuální osa a řízení rychlosti v prostředí NX

Jelikož se nepovedlo použít prvek *Převod* na virtuální osu, byl pro vytvoření převodu použit prvek *Elektronická váčka*. S pomocí tohoto prvku byla vytvořena lineární převodní charakteristika s převodovým poměrem mezi rotorem a hřídelí 25:1, tak jako u skutečného systému.



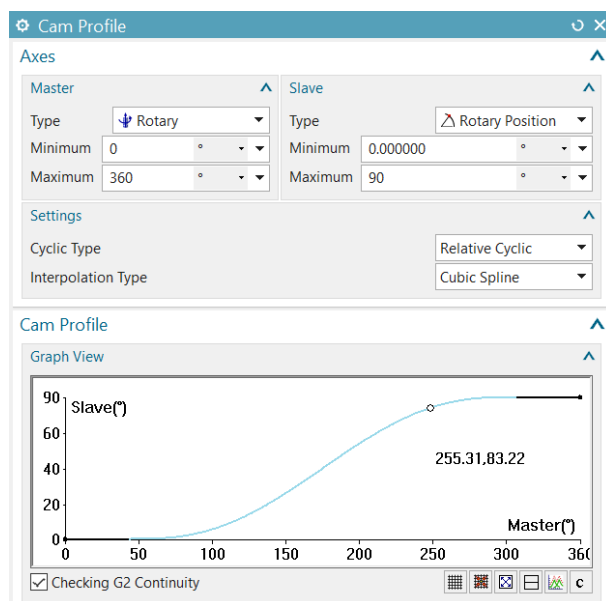
Obrázek 6.10: Přidání prvku elektronická váčka v prostředí NX





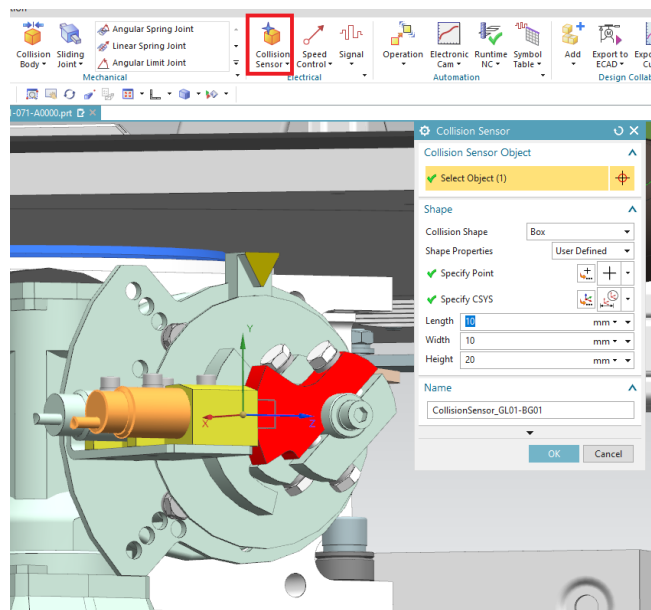
Obrázek 6.11: Převodní charakteristika mezi rotorem a hřídelí v prostředí NX

Druhá elektronická váčka byla vytvořena mezi hřídelí a karuselem. Pohyb karuselu je dán charakteristikou váčky. Karusel setrvává v pozici v případě, když je hřídel natočena v poloze mezi  $-45^\circ$  až  $+45^\circ$ . Na otočném stole se nachází 4 stanice a to znamená, že převodový poměr mezi hřídelí a karuselem bude 4:1.



Obrázek 6.12: Převodní charakteristika mezi hřídelí a karuselem v prostředí NX

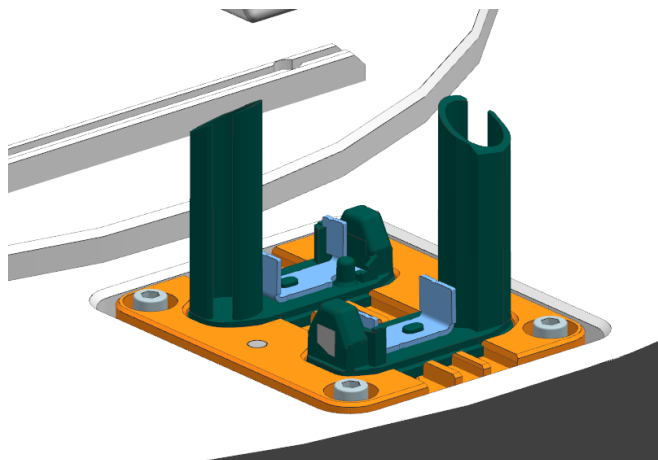
Na hřídeli se nachází díl, který je detekován senzorem v případě, že má dojít k zastavení motoru. Pro nastavení senzoru je vybrán z *Pásu karet* prvek *Kolizní senzor*, poté je označen daný senzor a vybrána oblast, která v případě detekce kolizního tělesa aktivuje senzor.



Obrázek 6.13: Přidání kolizního senzoru v prostředí NX

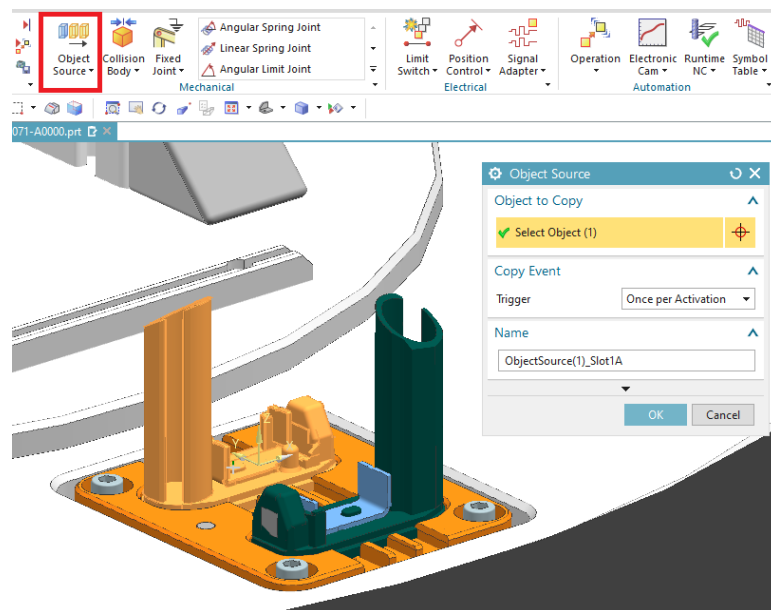
## 6.3 Materiál

Do modelu byly vloženy dva kusy materiálu do základací pozici lůžka na karuseli.



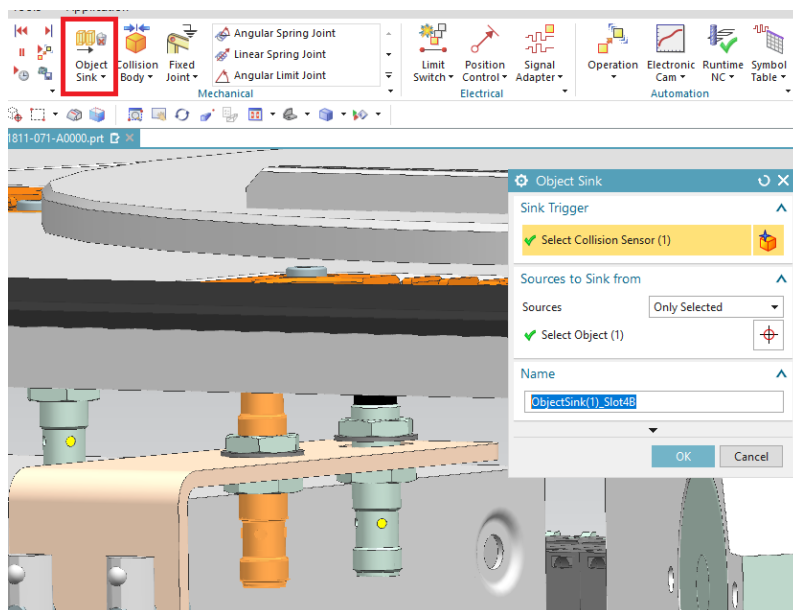
Obrázek 6.14: Lůžko se založeným materiálem v prostředí NX

Pro možnost vytvoření několika instancí materiálu je na materiál aplikován prvek *Zdroj objektů*, který zajistí vytváření dalších instancí podle času nebo aktivace. V tomto případě dojde k vytvoření instance materiálu po aktivaci spouštěcího signálu.



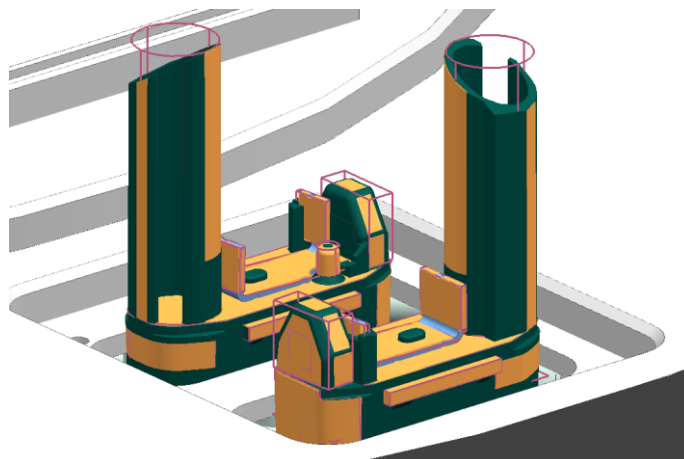
Obrázek 6.15: Přidání zdroje objektů v prostředí NX

Opakem *Zdroje objektu* je *Zánik objektu*. Při detekci vybraným senzorem instance materiálu zanikne. Materiál nezanikne při detekci senzorem, pokud je *Zánik objektu* deaktivován např. externím signálem.



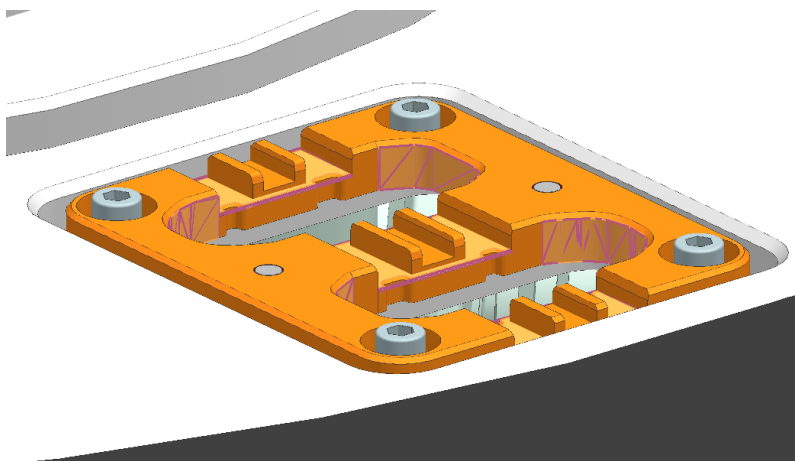
Obrázek 6.16: Přidání zániku objektu v prostředí NX

Na materiál bylo vytvořeno několik kolizních těles udržující jednoduchý tvar. Pokud by bylo použito kolizní těleso na celý povrch senzorů ve tvaru mesh simulace by byla zbytečně náročná.



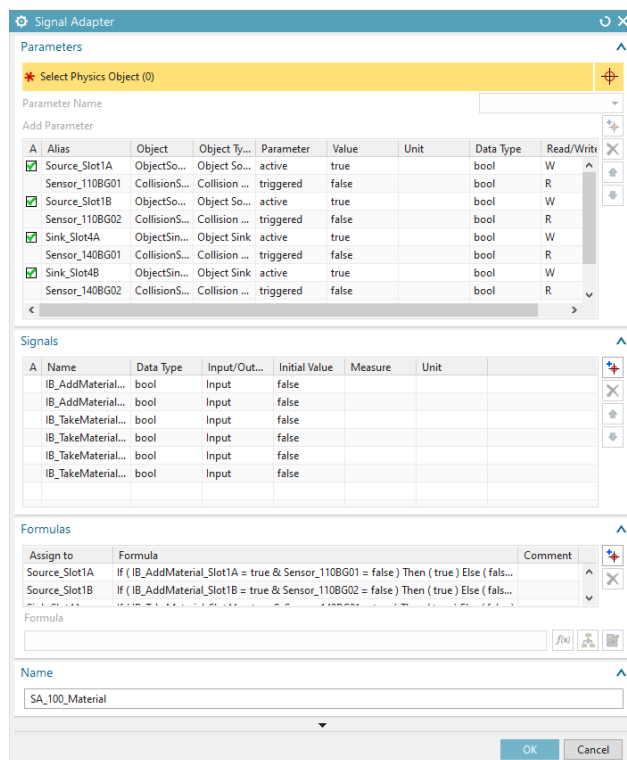
Obrázek 6.17: *Přidání kolizních těles na materiál v prostředí NX*

U všech 4 lůžek a NOK lůžka byly použita jednoduchá kolizní tělesa zabráňující propadnutí materiálu skrz díl.



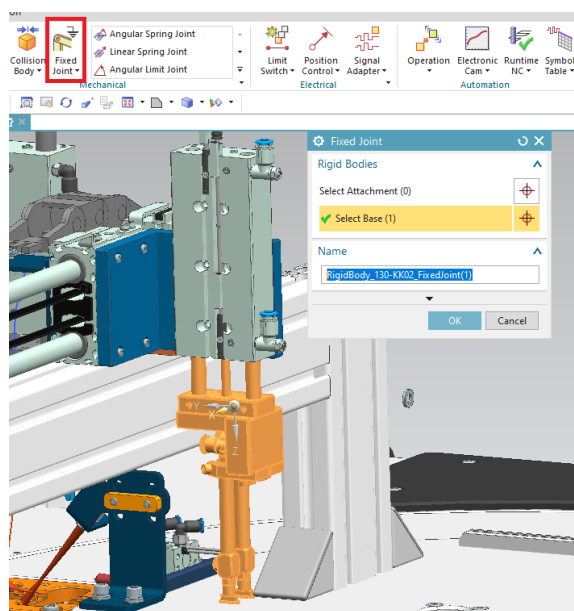
Obrázek 6.18: *Přidání kolizních těles na lůžka v prostředí NX*

Pro přidání materiálu byl vytvořen adaptér signálů. Pro vložení materiálu do základací pozice musí být senzor pro detekci přítomnosti dílu v log. 0 a externí signál pro přidání materiálu aktivní. Bez kontroly přítomnosti senzoru by došlo k vytvoření několika instancí, dokud by signál nebyl zase deaktivován. Pro odebrání materiálu je použita stejná podmínka. Akce jsou tedy provedeny pouze pokud příslušný senzor detekuje instanci materiálu.



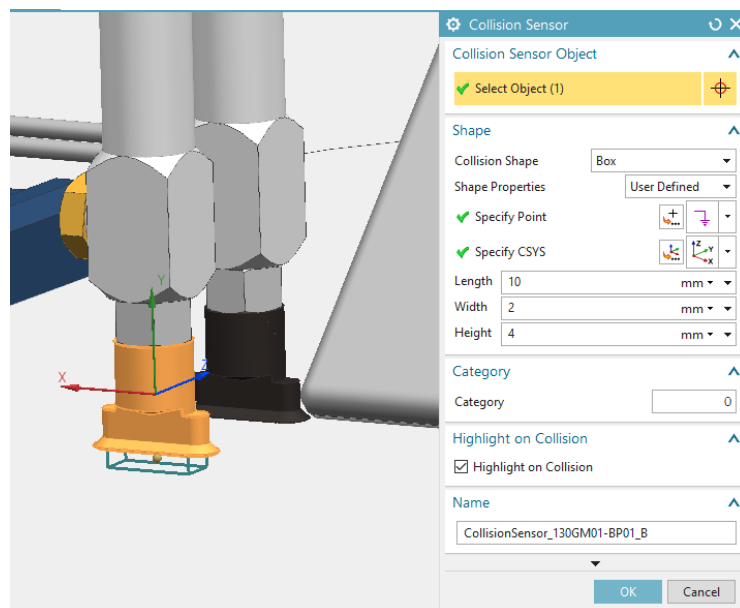
Obrázek 6.19: Přidání adaptéru signálu pro generování a odstraňování materiálu v prostředí NX

Z důvodu simulace přísátí materiálu vakuem bylo potřeba vytvořit pevný spoj, kde je základnou nedeformovatelné těleso s přísavkami. Jako přípevněný objekt bude během simulace přiřazena konkrétní instance materiálu.



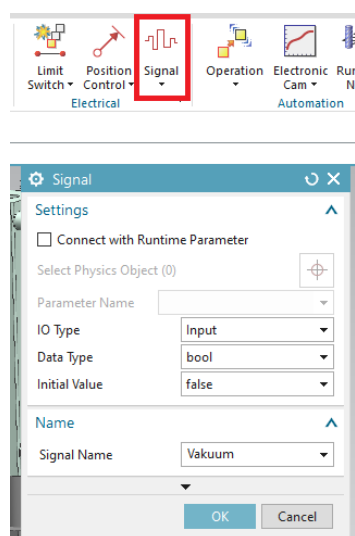
Obrázek 6.20: Přidání pevného spoje na manipulátor v prostředí NX

Na konci obou přísavek byly vytvořeny kolizní senzory pro detekci materiálu pod přísavkou. Kolizní senzory byly využity jako podmínka pro přípevnění instance materiálu.



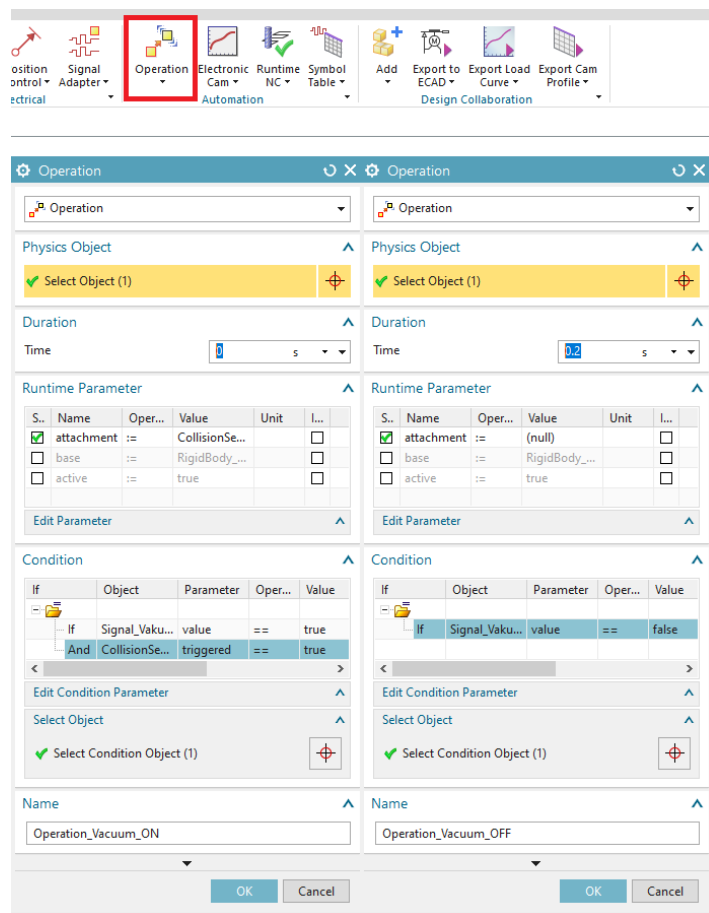
Obrázek 6.21: Přidání kolizních senzorů na přísavky v prostředí NX

Za účelem aktivování vakua bylo potřeba vytvořit signál, po jehož aktivaci dojde k přísátí materiálu.



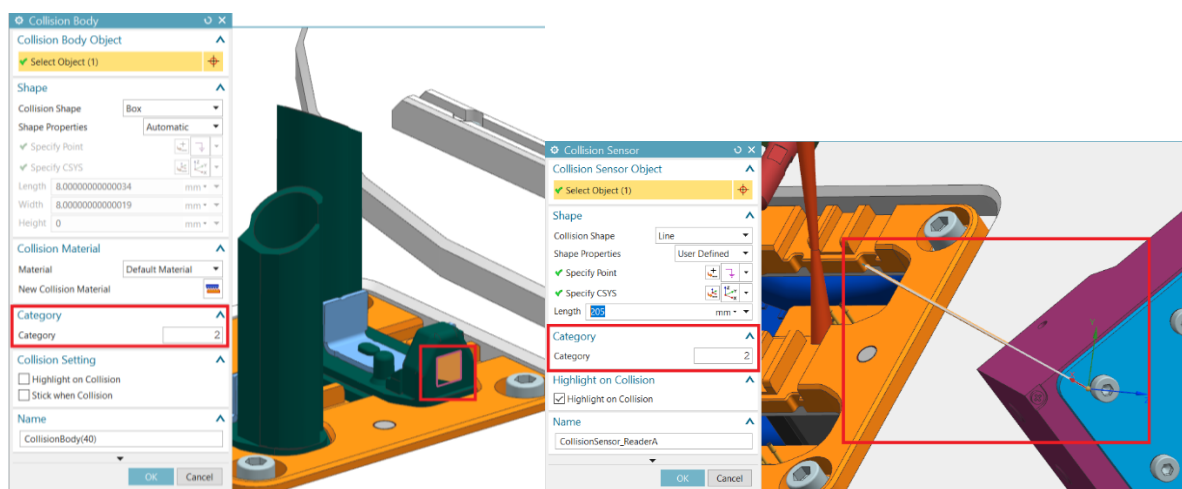
Obrázek 6.22: Vytvoření signálu pro aktivaci vakua v prostředí NX

V této části byla vytvořena *Operace*, přičemž objektem je pevný spoj a jako parametr bylo vybráno připevnění. Podmínkou připevnění je vytvořený signál v log. 1 a zároveň aktivní kolizní senzory na obou přísavkách. Pro uvolnění materiálu byla vytvořena druhá *Operace*, kde po deaktivaci signálu dojde k odebrání vazby připevnění a materiál je tak uvolněn.

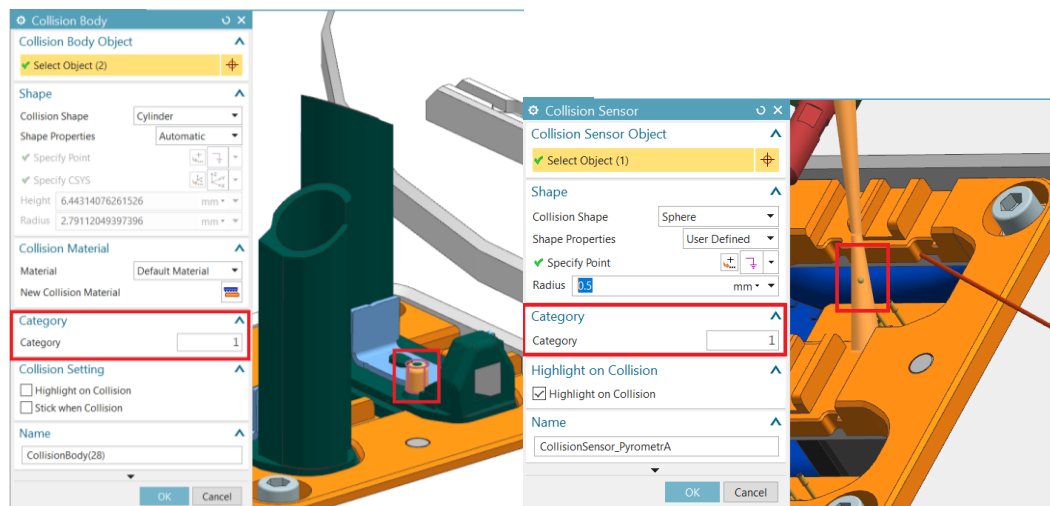


Obrázek 6.23: Vytvoření operací pro uchycení/uvolnění materiálu k manipulátoru v prostředí NX

Za účelem vyhodnocení, zda může měření dílu skončit jako OK, byly vytvořeny 2 kolizní senzory ve stanici 120. Senzory detekují jen konkrétní kolizní objekt s daným číslem kategorie. Kolizní senzory byly vytvořeny pro paprsek pyrometru a čtečku 2D kódu na obou slotech lůžka. V případě, že jeden z kolizních senzorů nebude aktivní, znamená to, že materiál není na správné pozici a měření musí být vyhodnoceno jako NOK.



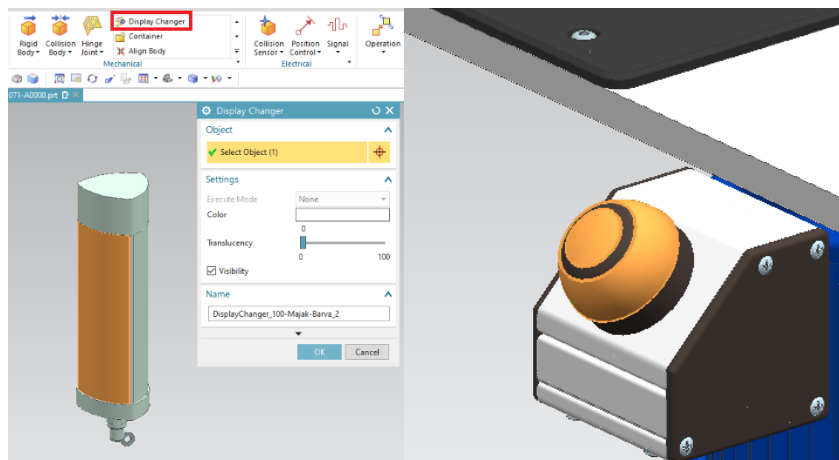
Obrázek 6.24: Nastavení kolizního tělesa a kolizního senzoru – čtečka 2D kódů



Obrázek 6.25: Nastavení kolizního tělesa a kolizního senzoru – pyrometr

## 6.4 Ostatní zařízení

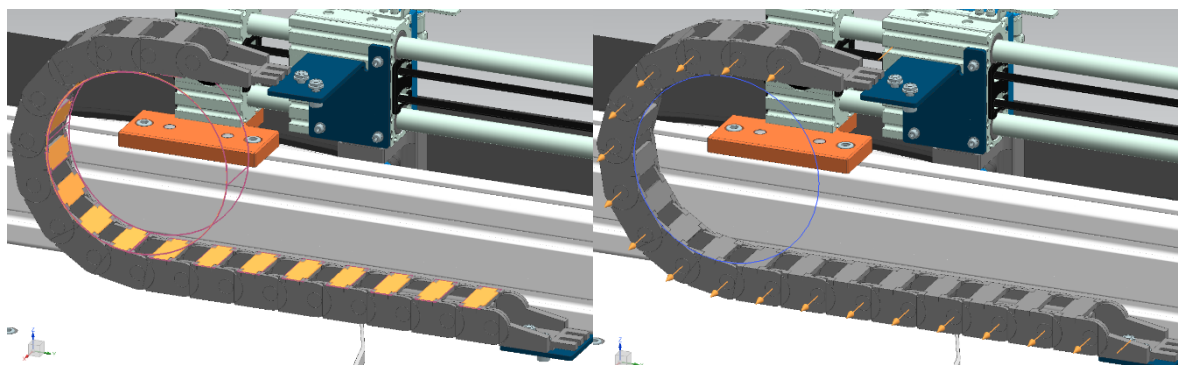
V prostředí NX byly simulovány také vizualizační zařízení jako je maják a potvrzovací tlačítko. Pro změnu zobrazení byl použit prvek *Měnič zobrazení*. Barva objektu je změněna v závislosti na externích signálech. První signál je celočíselného datového typu a určuje barvu objektu, druhý signál určuje, zda bude barva změna.



Obrázek 6.26: Přidání měniče zobrazení na maják a tlačítko v prostředí NX

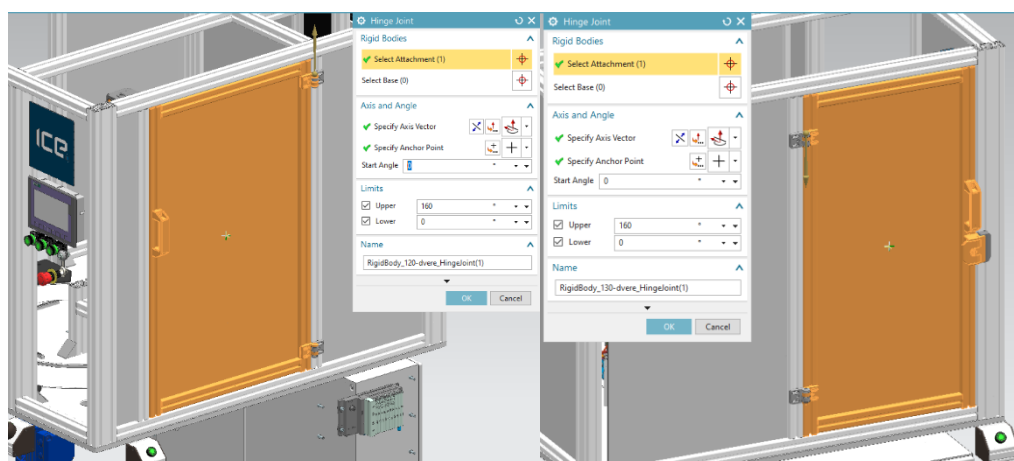
Pro vizualizaci pohybu energetického řetězu připevněného na manipulátoru bylo vytvořeno neviditelné kolizní těleso válcového tvaru. Na části řetězu byla vytvořena kolizní tělesa, která společně s kolizním tělesem válcového tvaru zajišťují udržení tvaru řetězu. Na jednotlivé klouby řetězu byl aplikován prvek *Otočný spoj*. Pro následování pohybu manipulátoru v ose X byl použit prvek *Elektronická vačka* s lineární charakteristikou a poměrem pohybu mezi manipulátorem a řetězem 2:1.





Obrázek 6.27: *Simulace pohybu energetického řetězu v prostředí NX*

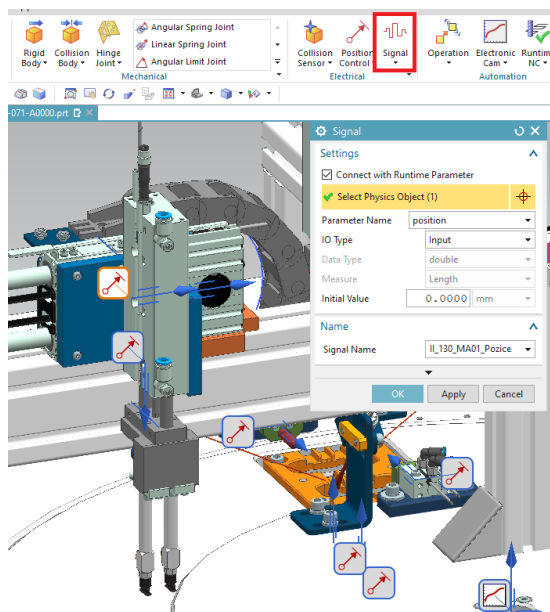
Simulace otvírání dveří byla vytvořena pomocí prvku *Otočný spoj* a *Řízení polohy*. Poloha dveří je určena externím signálem.



Obrázek 6.28: *Simulace pohybu dveří v prostředí NX*

## 6.5 Signály

Pro budoucí napojení akčních členů a senzorů s externím nástrojem bylo zapotřebí vytvořit signály a ty k těmto prvkům přiřadit. Signály mohou být vstupní nebo výstupní a datového typu bool, int nebo real. Pro vytvoření signálu, jehož hodnota vychází z libovolné rovnice, např. logický součin dvou proměnných nebo matematický přepočítání hodnoty, je zapotřebí použít adaptér signálu. Adaptér signálu lze také využít pro strukturované rozdělení signálů. Všechny signály jsou uloženy v symbolické tabulce. Signál, adaptér signálu nebo symbolická tabulka lze vytvořit přes *Pás karet*.



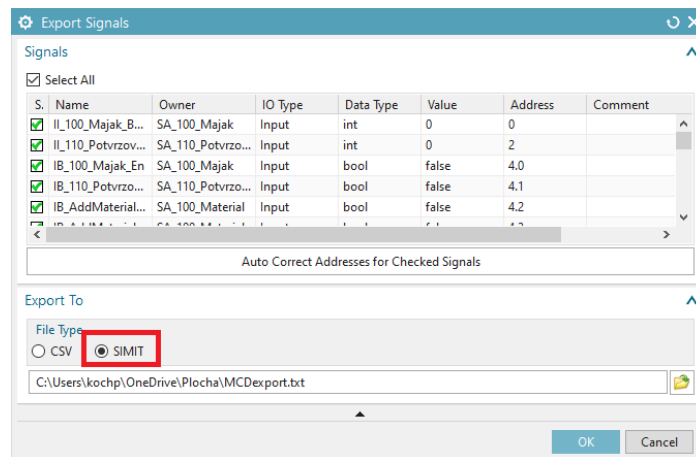
Obrázek 6.29: Přidání signálu na osu X manipulátoru v prostředí NX

Všechny signály byly pojmenovány s prefixem pro snadnější orientaci. První písmeno označuje vstup nebo výstup z pohledu NX MCD a druhé datový typ. Signály byly strukturovaně rozděleny do adaptéru signálu pro větší přehlednost. Nachází se zde také dvě symbolické tabulky pro vstupní a výstupní signály do nástroje SIMIT.

Signals		
+	SA_100_Dvere	Signal Adapter
+	SA_100_Majak	Signal Adapter
+	SA_100_Material	Signal Adapter
+	SA_100_Pohony	Signal Adapter
+	SA_100_Senzory	Signal Adapter
+	SA_110_PotvrzovacíTlacitko	Signal Adapter
+	SA_110_Senzory	Signal Adapter
	OB_110_BG01	Signal
	OB_110_BG02	Signal
+	SA_120_Pohony	Signal Adapter
	II_120_KK01_Pozice	Signal
	II_120_KK02A_Pozice	Signal
	II_120_KK02B_Pozice	Signal
	II_120_KK03_Pozice	Signal
	II_120_KK04_Pozice	Signal
+	SA_120_Senzory	Signal Adapter
+	SA_130_Pohony	Signal Adapter
+	SA_130_Senzory	Signal Adapter
+	SA_140_Senzory	Signal Adapter
	SIMIT_IN	Symbol Table
	SIMIT_OUT	Symbol Table

Obrázek 6.30: Seznam vytvořených signálů v prostředí NX

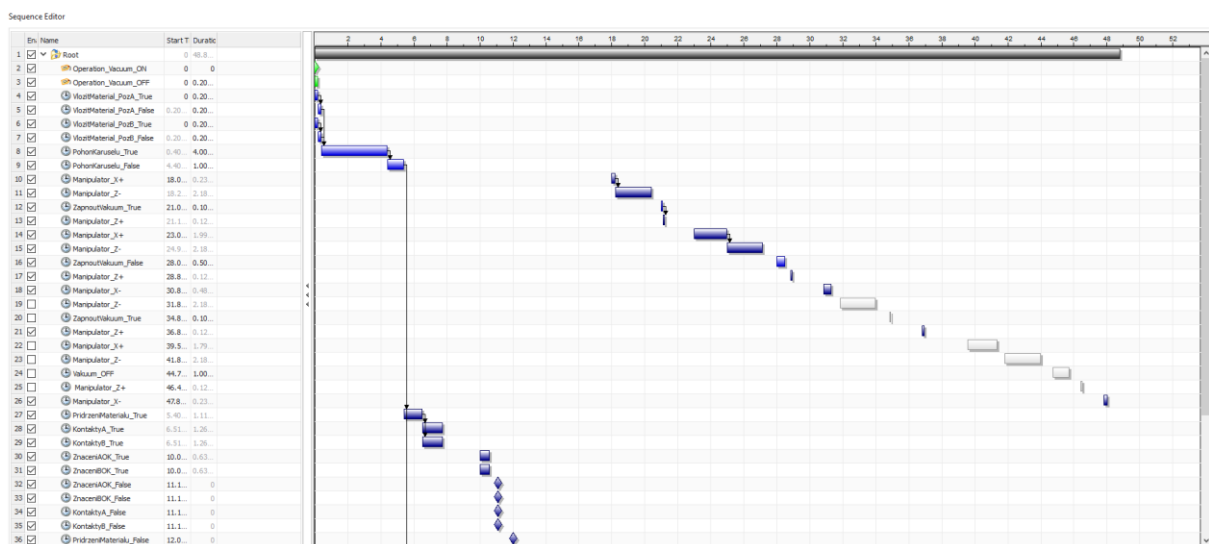
Export signálů byl proveden velmi snadno. Pravým klikem na signály a zvolením volby export bylo vyvoláno dialogové okno. V okně bylo potřeba vybrat dané signály, zaškrtnout volbu SIMIT a vybrat cestu k uložení souboru. Napojení signálu je popsáno v závěrečné kapitole práce viz. 9.2.



Obrázek 6.31: Export signálů z prostředí NX

## 6.6 Time-based a Event-based simulace

V prostředí je možné vytvořit časově závislé sekvence vytvořených operací. Při zahájení simulace je do lůžka vložen materiál, v daný časový okamžik je posunut karusel a za 4 vteřiny je vysunut přídržovací válec atd. Touto jednoduchou simulací lze včas odhalit chyby a otestovat funkčnost vytvořeného modelu. Druhým typem simulace v NX je event-based simulace, kde je použit stejný sekvenční editor, ale operace jsou vykonávány na základě událostí. Jako událost je považována např. přechod stavu senzoru z log. 0 do log. 1. Tyto již vytvořené sekvence lze také exportovat a použít pro vytváření nebo vygenerování řídicího programu. Po přiřazení všech prvků a vytvoření signálu bylo využito time-based simulace pro ověření funkčnosti vytvořeného simulačního modelu.



Obrázek 6.32: Ganttův diagram pro vytváření sekvencí v prostředí NX

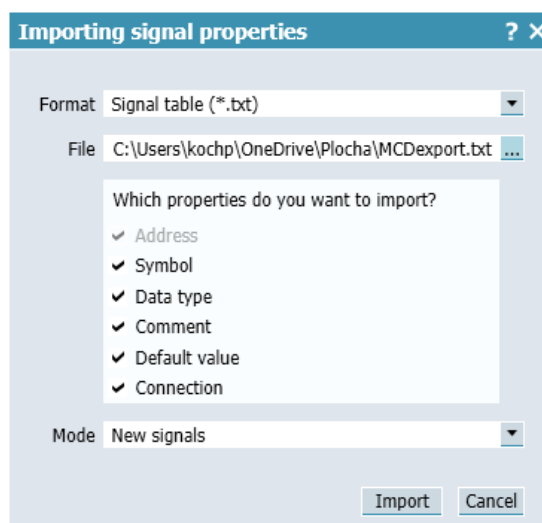
## 7. Realizace elektrického a behaviorálního modelu

V nástroji SIMIT bylo vytvořeno spojení SHM a exportována tabulka, i když zde zatím žádné signály nejsou. Je to z důvodu získání hlavičky pro následný import signálů. První řádek exportovaných signálů ze SIMIT je přkopírován na první řádek souboru s exportovanými signály z NX MCD. Nastavení komunikace s NX MCD a komunikační aplikací je popsáno v závěrečné kapitole práce, viz. 9.1.



Obrázek 7.1: *Modifikace hlavičky exportovaných signálů z NX MCD*

Import signálu je poté proveden v sekci spojení, v daném případě SHM. Je vybrán exportovaný soubor z NX MCD s modifikovanou hlavičkou a stisknuto tlačítko import.



Obrázek 7.2: *Import signálů z NX do SIMIT*

Seznam importovaných signálů je pak zobrazen v tabulce společně s adresami a datovým typem daného signálu. Všechny signály mají adresu začínající písmenem M, jelikož se jedná o sdílenou paměť, a ne o fyzicky vstupní nebo výstupní signál.

MCDSHM (SHM)				
Inputs				
Default	Symbol name	Address	Data type	Comment
0	II_100_Majak_Barva	MW0	INT	
0	II_110_PotvrzovacíTlacitko_Barva	MW2	INT	
<input type="checkbox"/>	IB_100_Majak_En	M4.0	BOOL	
<input type="checkbox"/>	IB_110_PotvrzovacíTlacitko_En	M4.1	BOOL	
<input type="checkbox"/>	IB_AddMaterial_Slot1A	M4.2	BOOL	
<input type="checkbox"/>	IB_AddMaterial_Slot1B	M4.3	BOOL	
<input type="checkbox"/>	IB_TakeMaterial_Slot4A	M4.4	BOOL	
<input type="checkbox"/>	IB_TakeMaterial_Slot4B	M4.5	BOOL	
<input type="checkbox"/>	IB_TakeMaterial_Slot1A	M4.6	BOOL	
<input type="checkbox"/>	IB_TakeMaterial_Slot1B	M4.7	BOOL	
0	II_120_KK01_Pozice	MW5	INT	
0	II_120_KK02_Pozice	MW7	INT	
Outputs				
	Symbol name	Address	Data type	Comment
	OB_110_BG01	M20.0	BOOL	
	OB_110_BG02	M20.1	BOOL	
	OB_130_BG01	M20.2	BOOL	
	OB_130_BG02	M20.3	BOOL	
	OB_140_BG01	M20.4	BOOL	
	OB_140_BG02	M20.5	BOOL	
	OB_130_BP01	M20.6	BOOL	
	OB130_CL01_BG01	M20.7	BOOL	
	OB130_CL01_BG02	M23.0	BOOL	
	OB130_CL01_BG03	M23.1	BOOL	
	OB130_CL01_BG04	M23.2	BOOL	
	OB130_CL01_BG05	M23.3	BOOL	

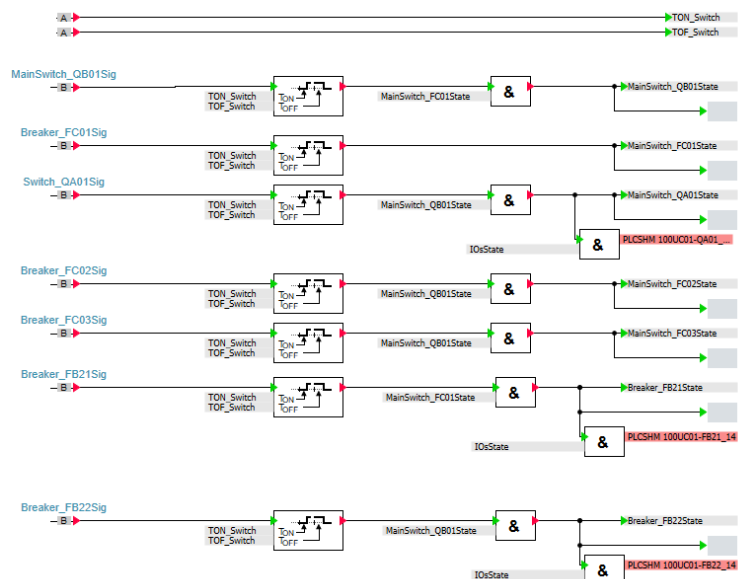
Obrázek 7.3: Importované signály z NX v SIMIT

V projektu SIMIT byl použit asynchronní operační mód. Ideální volbou by byl mód synchronní sběrnice, který je nejvhodnější pro real-time simulace. Bohužel tento mód není součástí SIMIT V9.0. Periody výpočtu simulačních modelů v SIMIT byly použity od 10 ms po 100 ms. *Time slice 7* a *Time slice 8* byly upraven pro rychlejší výpočet požadovaných modelů, zbytek nastavení period zůstal původní.

Time slice 1 [ms]	50
Time slice 2 [ms]	100
Time slice 3 [ms]	200
Time slice 4 [ms]	400
Time slice 5 [ms]	800
Time slice 6 [ms]	1600
Time slice 7 [ms]	10
Time slice 8 [ms]	20
Operating mode	Asynchronous

Obrázek 7.4: Nastavení simulace v SIMIT

## 7.1 Diagramy napájení a prvků rozvaděče

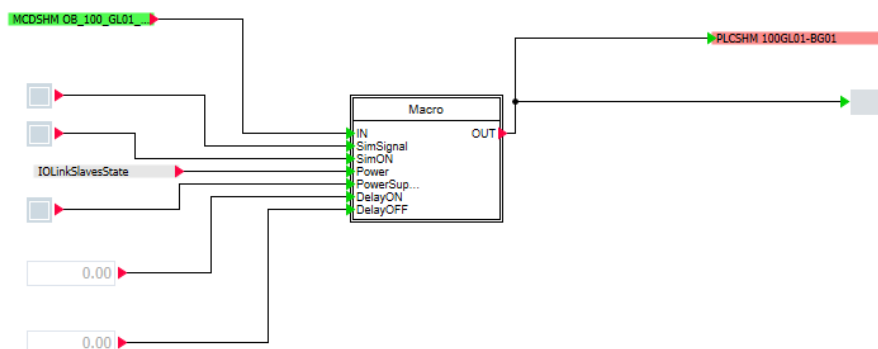


Obrázek 7.5: Část diagramu simulující prvky rozvaděče v SIMIT

Pro simulaci prvků rozvaděče bylo využito zpoždění signálů simulující nahození jističe, přepnutí vypínače atd. Pro dodání elektrické energie do daného zařízení byly využity logické funkce, které byly použity na základě elektro-dokumentace. Zařízení v případě odpojení napájení přestává fungovat. Byly zde napojeny vstupní signály pro PLC, jako jsou stavy konkrétních jističů, senzory tlaku vzduchu, stavy kontaktů relé atd.

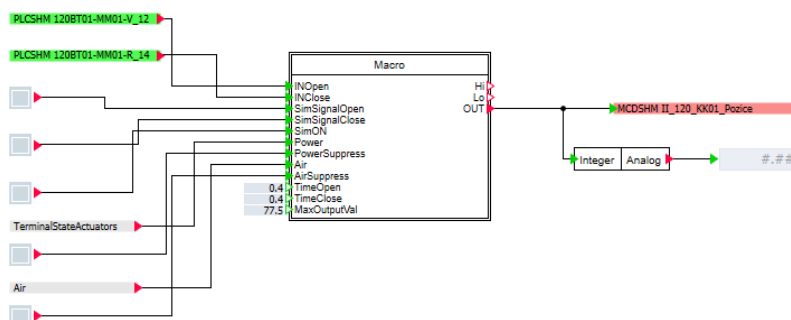
## 7.2 Diagramy senzorů a ventilů

Pro senzory a řízení polohy válce byly v SIMIT vytvořeny makra. Makro pro senzory obsahuje vstupní signály jako je aktuální stav senzoru z NX, simulační signály, signál určující, zda je senzor napájen a zpoždění stavu senzoru. Na výstupu makra se nachází konkrétní stav senzoru pro PLC po zpracování vnitřní logikou.



Obrázek 7.6: Diagram senzoru v SIMIT

Makro pro ventil obsahuje vstupní signály z PLC pro otevření a zavření ventilu, simulační signály, signál napájení, signál dostatečného tlaku vzduchu, doba vysunutí a zasunutí válce do konečné pozice a maximální výstupní hodnota na výstupu. Výstupem makra je pozice válce pro NX.



Obrázek 7.7: Diagram ventilu v SIMIT

## 7.3 Diagramy pohonů

### 7.3.1 Motor karuselu

Pro otáčení stolu slouží motor SEW DFR63L4. Motor je umístěn pod otočným stolem a je připojen přes šnekovou převodovku na válcovou vačku, která krokově otáčí stolem v záběru s axiálně uloženými kladkami. V systému je použita šneková převodovka Tramec SFK 40 R 25 s převodovým poměrem 25:1. Otočný stůl obsahuje 4 stanice a jeho celkový moment setrvačnosti je  $1,8 \text{ kg} \cdot \text{m}^2$ . Celková hmotnost zátěže motoru je 21 kg. Čas pro přejetí do další stanice je uveden 0,94 vteřin a je označen jako čas přestavby. Pozice karuselu vůči pozici rotoru byla řešena v NX MCD. Níže je vypočtena doba dosažení maximálních otáček motoru pomocí zjednodušeného výpočtu.

Znamé parametry:

[66; 67]

$P_N = 0,25 \text{ kW}$ ,  $M_N = 1,8 \text{ Nm}$ ,  $n_N = 1300 \text{ ot} \cdot \text{min}^{-1}$ ,  $M_H/M_N = 1,7$ ,  $M_{MAX} = 3,2 \text{ Nm}$ ,  $J_M = 5,6 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$ ,  $J_G = 0,262 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$ ,  $i_{MG} = 25$ ,  $M_L = 0 \text{ Nm}$ ,  $J_L = 1,8 \text{ kg} \cdot \text{m}^2$

Výpočet potřebných parametrů:

$$M_N = \frac{P_N}{\frac{n_N}{60} \cdot 2\pi} = \frac{250}{\frac{1300}{60} \cdot 2\pi} = 1,837 \text{ Nm} \quad (1)$$

$$M_H = 1,7 \cdot M_N = 1,7 \cdot 1,837 = 3,123 \text{ Nm} \quad (2)$$

$$M_B = M_H - 2 \cdot M_L = 3,123 - 2 \cdot 0 = 3,123 \text{ Nm} \quad (3)$$

Doba rozjetí motoru:

[68]

$$t_A = \frac{\left(J_M + J_G + \frac{J_L}{i^2}\right) \cdot \frac{n_N}{60} \cdot 2\pi}{(M_H - M_L)} \quad (4)$$

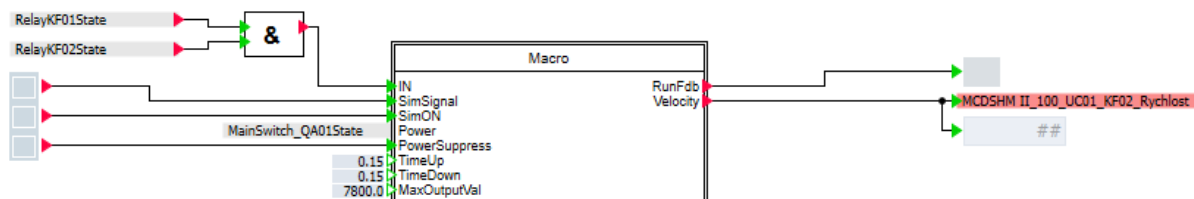
$$t_A = \frac{\left(5,6 \cdot 10^{-4} + 0,262 \cdot 10^{-4} + \frac{1,8}{25^2}\right) \cdot \frac{1300}{60} \cdot 2\pi}{(3,123 - 0)} = 0,15 \text{ s}$$

Doba zastavení motoru:

[68]

$$t_B = \frac{\left(J_M + J_G + \frac{J_L}{i^2}\right) \cdot \frac{n_N}{60} \cdot 2\pi}{(M_B - M_L)}$$

$$t_B = \frac{\left(5,6 \cdot 10^{-4} + 0,262 \cdot 10^{-4} + \frac{1,8}{25^2}\right) \cdot \frac{1300}{60} \cdot 2\pi}{(3,123 - 0)} = 0,15 \text{ s} \quad (5)$$

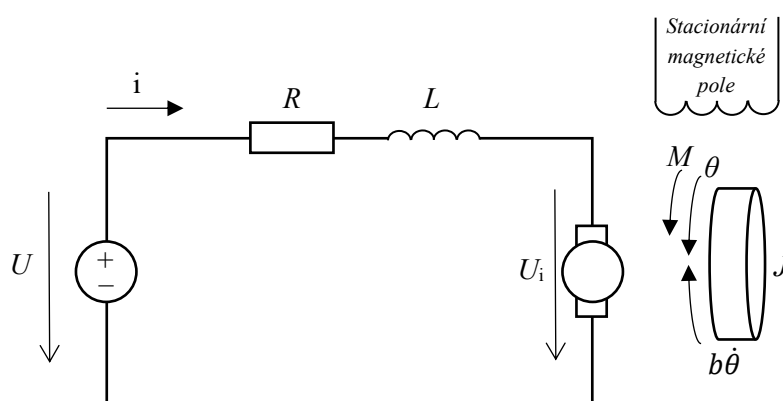


Obrázek 7.8: Diagram motoru karuselu v SIMIT

Diagram je vytvořen s pomocí makra, které obsahuje vstupní a výstupní signály. V případě splnění podmínek spuštění jsou otáčky motoru zvyšovány podle křivky dané komponentou *Motor* z knihovny *Drives*. Pro spuštění motoru musí být sepnuty kontakty stykačů KF01 a KF02. Jsou zde také simulační signály pro jednoduché ovládání rovnou z diagramu. Na vstupní signál *Power* je připojen signál určující, zda je do zařízení přivedena elektrická energie. Na vstupech *TimeUp* a *TimeDown* je vypočtena hodnota rozjetí a zastavení motoru 0,15 vteřin. Hodnota *MaxOutputVal* označuje maximální možnou rychlost. NX očekává hodnotu rychlosti ve %s proto je tato hodnota nastavena na 7800. Na výstupu je napojen signál rychlosti motoru pro fyzikálně-kinematický model.

### 7.3.2 Lineární DC motor

Jako regulovaná soustava byl vytvořen model DC motoru. Stejnoseměrný motor lze vyjádřit náhradním schématem zapojením s rezistorem a cívkou v sérii.



Obrázek 7.9: Náhradní schéma DC motoru [69]

Pro stejnosměrné motory platí momentová a napěťová rovnice. Na vinutí kotvy je indukováno napětí v případě, že rotor rotuje, což je dáno úhlovou rychlostí kotvy  $\omega$  a prochází magnetický tokem  $\Phi$ . Na rotoru vzniká moment  $M$  v závislosti na proudu vinutím  $i$ .  $K_e$  označuje konstantu elektromotorického napětí a  $K_t$  konstantu momentu motoru.



$$U_i = K_e \cdot \Phi \cdot \omega = K_e \cdot \Phi \cdot \frac{d\theta}{dt} \quad (6)$$

$$M = K_t \cdot \Phi \cdot i \quad (7)$$

Po aplikaci druhého Kirchhoffova zákona je získána první rovnice (8). Do rovnice je dosazena předchozí rovnice (6).

$$U = U_R + U_L + U_i$$

$$U = R \cdot i + L \frac{di}{dt} + K_e \cdot \Phi \cdot \frac{d\theta}{dt} \quad (8)$$

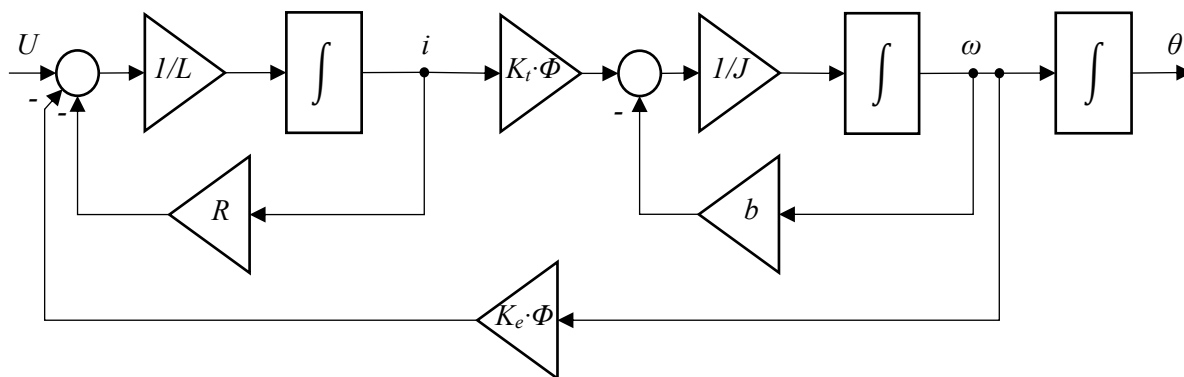
$$\frac{di}{dt} = \frac{1}{L} (U - R \cdot i - K_e \cdot \Phi \cdot \frac{d\theta}{dt})$$

Použitím Newtonova zákona o zachování hybnosti vznikne momentová rovnice ve tvaru níže (9), kde  $b$  je třecí konstanta motoru a  $J$  je moment setrvačnosti. Do rovnice byla následně dosazena rovnice (7).

$$J \frac{d^2\theta}{dt^2} = M - b \frac{d\theta}{dt}$$

$$\frac{d^2\theta}{dt^2} = \frac{1}{J} (K_t \cdot \Phi \cdot i - b \frac{d\theta}{dt}) \quad (9)$$

Pomocí výsledných rovnic (8) a (9) byl vytvořen model stejnosměrného motoru v SIMIT. Pro větší přehlednost bylo zapojení modelu překresleno. V modelu je předpokládána konstantní hodnota magnetického pole.



Obrázek 7.10: Model DC motoru

Model soustavy byl vytvořen jako makro obsahující řídicí vstupní napětí, které je přivedeno z regulátoru proudu. Další vstupy obsahují parametry modelu a mezní výstupní hodnoty pozice, rychlosti a proudu. Výstupem jsou aktuální hodnoty pozice, rychlosti a proudu. Diagram také obsahuje matematické přepočty pro NX a PLC.

Protože soustava počítá s rychlostí v jednotkách úhlové rychlosti  $\text{rad} \cdot \text{s}^{-1}$ , byla přepočtena hodnota maximální rychlosti motoru lineárního pohonu. Maximální rychlost motoru lineární osy X je

2720 ot·min<sup>-1</sup>. Posuvová konstanta je 58 mm·ot<sup>-1</sup>. Maximální možná pozice otočení motoru byla také přepočtena, přičemž maximální pozice lineárního pohybu je 406 mm. [70; 71]

Výpočet maximální úhlové rychlosti motoru:

$$\omega_{\max} = \frac{n_{\max}}{\text{posuvová konst.}} \cdot 2\pi$$

$$\omega_{\max} = \frac{2720}{58 \cdot 10^{-3}} \cdot 2\pi = 284,84 \text{ rad} \cdot \text{s}^{-1} \quad (10)$$

Výpočet maximální úhlové dráhy motoru:

$$\theta_{\max} = \frac{s_{\max}}{\text{posuvová konst.}} \cdot 2\pi$$

$$\theta_{\max} = \frac{406 \cdot 10^{-3}}{58 \cdot 10^{-3}} \cdot 2\pi = 43,98 \text{ rad} \quad (11)$$

V dokumentaci k pohonu s lineárním řemenem ELGR-TB-35-400-0H-ST-E-AD-RL-2.5E a motoru lineární osy EMMS-ST-57-S-SE-G2 byla nalezena a vypočtena část potřebných parametrů pro vytvoření modelu motoru. Parametry, které v dokumentaci nebyly uvedeny byly experimentálně odhadnuty za pomoci zkoumání modelu v prostředí MATLAB Simulink.

Parametry motoru:

$$U = 48 \text{ V}, I_N = 5 \text{ A}, n_{\max} = 2720 \text{ ot} \cdot \text{min}^{-1}, L = 0,5 \text{ mH}, R = 0,15 \Omega \quad [71]$$

Parametry pohonu:

$$J_O = 40,26 \text{ kg} \cdot \text{mm}^2, k = 0, J_W = 36,75 \text{ kg} \cdot \text{mm}^2, J_S = 0,26 \text{ kg} \cdot \text{mm}^2 \cdot \text{m}^{-1}, l = 0,4 \text{ m},$$

$$J_L = 85,19 \text{ kg} \cdot \text{mm}^2 \cdot \text{kg}^{-1}, m = 3,05 \text{ kg} \quad [70]$$

$$J = J_O + k \cdot J_W + J_S \cdot l + J_L \cdot m$$

$$J = 10^{-6} \cdot (40,26 + 0 \cdot 36,75 + 0,26 \cdot 0,4 + 85,19 \cdot 3,05) \quad (12)$$

$$J = 300,1935 \cdot 10^{-6} \text{ kg} \cdot \text{m}^2$$

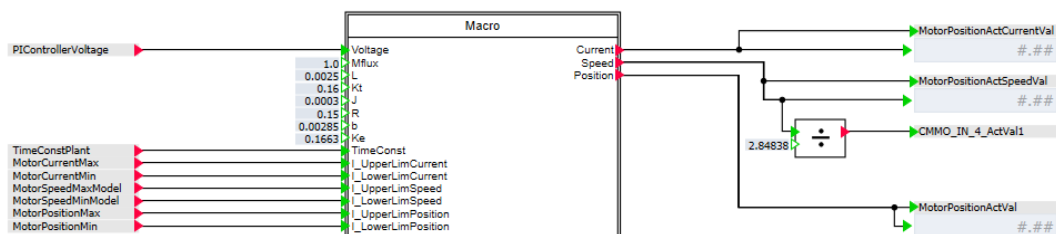
Zvolené parametry:

$$K_t = 0,16 \text{ Nm} \cdot \text{A}, K_e = 0,1663 \text{ V} \cdot \text{rad} \cdot \text{s}^{-1}, b = 2,85 \cdot 10^{-3} \text{ Nm} \cdot \text{s}, \Phi = 1 \text{ Wb}$$

Během simulací bylo zjištěno, že při těchto parametrech modelu dochází ke kmitání proudu o vysoké frekvenci a bez použití rychlého regulátoru by vznikl nestabilní systém. Protože regulátor v SIMIT je schopen nejrychleji vzorkovat s periodou 10 ms, byla hodnota indukivity vinutí zvýšena pětkrát. Po této úpravě byla regulace proudu v obvodu v nástroji SIMIT již realizovatelná.

Tabulka 7.1: Parametry modelu motoru

$L$ [H]	$R$ [ $\Omega$ ]	$J$ [ $\text{kg} \cdot \text{m}^2$ ]	$b$ [ $\text{Nm} \cdot \text{s}$ ]	$K_t$ [ $\text{Nm} \cdot \text{A}$ ]	$K_e$ [ $\text{V} \cdot \text{rad} \cdot \text{s}^{-1}$ ]
$2,5 \cdot 10^{-3}$	0,15	$300,1935 \cdot 10^{-6}$	$2,85 \cdot 10^{-3}$	0,16	0,1663



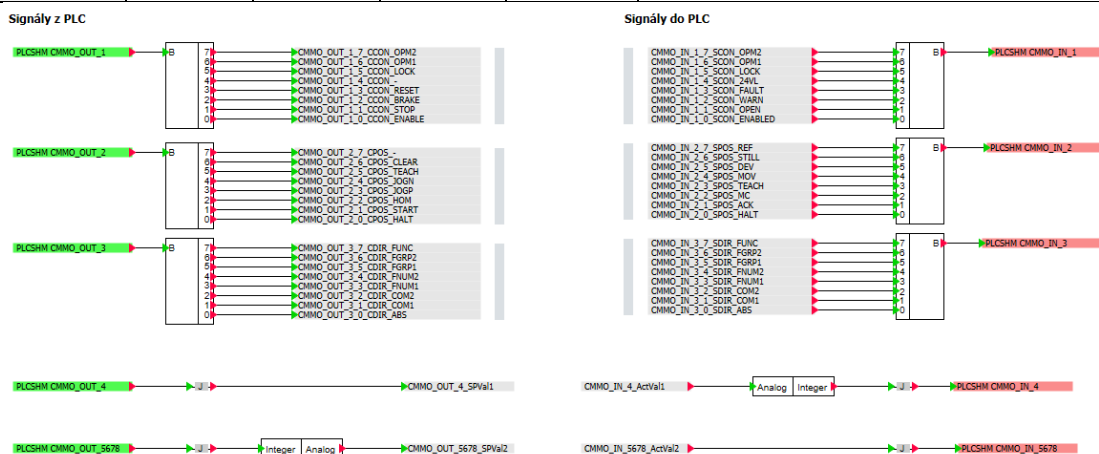
Obrázek 7.11: Diagram lineárního DC motoru manipulátoru v SIMIT

### 7.3.3 Řídící jednotka CMMO

CMMO jednotka pro komunikaci používá 8 vstupních a 8 výstupních bytů. Vstupní data jsou statusová a výstupní data příkazy (bráno z pohledu PLC). Model chování byl vytvořen pro komunikaci s jednotkou v FHPP operačním režimu Direct mode. Simulační model jednotky s PLC komunikuje téměř jako s opravdovou jednotkou. Signály jako je např. chyba může být v simulačním modelu jednoduše nasimulována a sledována následná reakce PLC. Chování jednotky na základě výstupních signálu z PLC bylo vytvořeno podle dokumentu výrobce, kde byly všechny různé stavy pochopitelně vysvětleny.

Tabulka 7.2: Seznam výměny signálu s jednotkou CMMO-ST [72]

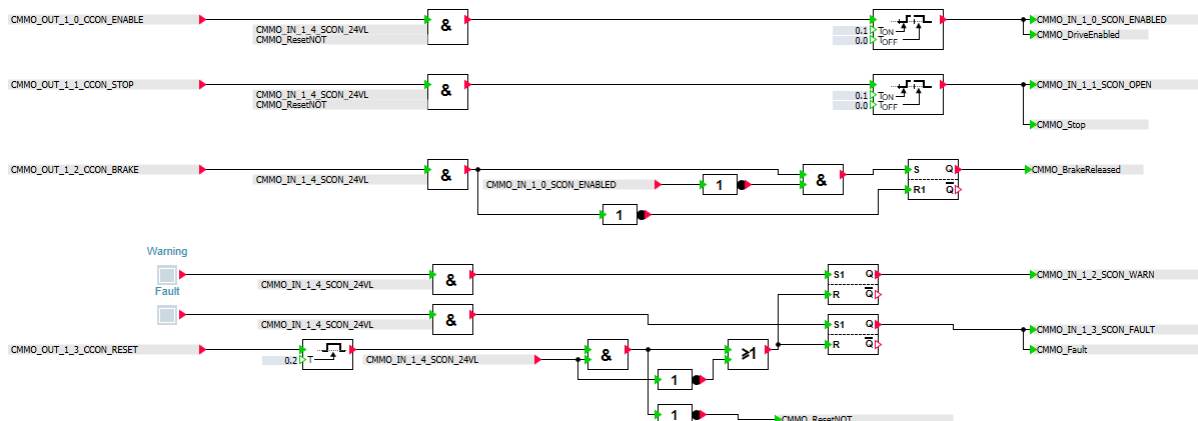
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
<b>Výstupní data</b>	CCON	CPOS	CDIR	Žádaná hodnota 1	Žádaná hodnota 2			
<b>Vstupní data</b>	SCON	SPOS	SDIR	Aktuální hodnota 1	Aktuální hodnota 2			



Obrázek 7.12: Zpracování I/O signálu pro CMMO jednotku v SIMIT

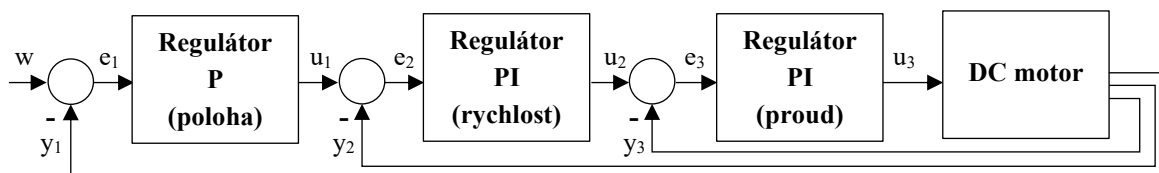
Ve výstupních datech CCON se nachází signály pro povolení pohonu, zastavení operace, uvolnění brzdy, reset, uzamčení jednotky a volba operačního módu. Signály SCON obsahují signály jako pohony povoleny, operace povolena, varování, chyba, napájení, stav uzamčení jednotky a zpětná vazba zvoleného operačního módu. Signály v CPOS obsahují pohybové instrukce např. homing, jogging nebo odstartování pohybové instrukce a SPOS následně stavy pohybu jako je pozastavení, pohyb dokončen nebo pohon referencován. V bytu CDIR se nachází signály pro volbu absolutního/relativního

pohybu, řídicí mód, funkční číslo, funkční skupina a funkce. SDIR obsahuje stejné signály kterými potvrzuje jejich zvolení. Chování bylo vytvořeno, tak že na bytu 4 se nachází žádaná a aktuální rychlost v procentech a na bytech 5 až 8 se nachází požadovaná a aktuální pozice manipulátoru v jednotkách mikrometrů.



Obrázek 7.13: Část logiky zpracování signálu CMMO jednotky v SIMIT

Pro řídicí jednotku byly také vytvořeny 3 regulátory pro kaskádovou regulaci manipulátoru. Společně se soustavou se jedná o rozvětvený regulační obvod s pomocnou regulovanou veličinou. Obvod se skládá z jednoho P regulátoru pro regulaci polohy, dvojice PI regulátorů pro regulaci rychlosti a proudu a regulované soustavy představující stejnosměrný motor.



Obrázek 7.14: Blokové schéma kaskádového regulačního obvodu

Pro regulátor bylo vytvořeno makro se vstupy pro regulační odchylku, konstantu proporcionální složky, konstantu integrační složky, časovou konstantu integrátoru, saturaci akčního zásahu a koeficient omezení integrační složky (anti-windup). Výstupem tohoto makra je akční zásah regulátoru. Maximální hodnota akčního zásahu regulátoru polohy představuje žádanou hodnotu regulátoru rychlosti a její omezení je procentuálně řízeno z PLC. Maximální rychlost motoru je vypočtena v části týkající se motoru manipulátoru. Maximální hodnota proudu tekoucího do motoru je omezena na 5,7 A. Motor je řízen jednotkou napětím v rozsahu  $\pm 24$  V. [73]



$$\theta(s) = \frac{K_t \cdot \Phi \cdot I(s)}{s(Js + b)} \quad (16)$$

Přenos soustavy s proudem na výstupu byl odvozen pomocí rovnic (13) a (16).

$$\begin{aligned} U(s) &= I(s) \cdot (R + Ls) + K_e \cdot \Phi \cdot s \cdot \frac{K_t \cdot \Phi \cdot I(s)}{s(Js + b)} \\ \frac{U(s)}{I(s)} &= (R + Ls) + \frac{K_e \cdot K_t \cdot \Phi^2}{(Js + b)} \\ G_I(s) &= \frac{I(s)}{U(s)} = \frac{1}{(R + Ls) + \frac{K_e \cdot K_t \cdot \Phi^2}{(Js + b)}} \end{aligned} \quad (17)$$

Přenos soustavy s rychlostí na výstupu byl odvozen pomocí rovnic (13) a (15).

$$\begin{aligned} U(s) &= \frac{s\theta(s) \cdot (Js + b)}{K_t \cdot \Phi} \cdot (R + Ls) + K_e \cdot \Phi \cdot s\theta(s) \\ \frac{U(s)}{s\theta(s)} \cdot K_t \cdot \Phi &= (Js + b) \cdot (R + Ls) + K_e \cdot K_t \cdot \Phi^2 \\ G_\omega(s) &= \frac{s\theta(s)}{U(s)} = \frac{K_t \cdot \Phi}{(Js + b) \cdot (R + Ls) + K_e \cdot K_t \cdot \Phi^2} \end{aligned} \quad (18)$$

Přenos soustavy s polohou na výstupu byl odvozen z rovnice (18).

$$G_\theta(s) = \frac{\theta(s)}{U(s)} = \frac{K_t \cdot \Phi}{(Js + b) \cdot (R + Ls) + K_e \cdot K_t \cdot \Phi^2} \cdot \frac{1}{s} \quad (19)$$

Přenos vnitřní smyčky pro regulátor rychlosti  $G_{IN1}$  byl popsán rovnicí (20) a přenos vnitřní smyčky pro regulátor polohy  $G_{IN2}$  rovnicí (21). Obě rovnice byly vytvořeny podle schématu kaskádního regulačního obvodu, viz. Obrázek 7.14. Přičemž  $G_{R3}$  je přenos regulátoru proudu,  $G_{R2}$  je přenos regulátoru rychlosti a  $G_{R1}$  je přenos regulátoru polohy.

$$G_{IN1}(s) = \frac{G_{R3}(s) \cdot G_\omega(s)}{1 + G_{R3}(s) \cdot G_I(s)} \quad (20)$$

$$G_{IN2}(s) = \frac{G_{R2}(s) \cdot \frac{G_{R3}(s) \cdot G_\theta(s)}{1 + G_{R3}(s) \cdot G_I(s)}}{1 + G_{R2}(s) \cdot G_{IN1}(s)} \quad (21)$$

Přenosy (17), (20) a (21), společně s parametry DC motoru, byly přepsány a následně vypočteny v MATLABu. Za pomoci nástroje PID Tool byly postupně navrženy jednotlivé regulátory. Nejdříve proběhl návrh regulátoru proudu, poté rychlosti a na závěr byl navržen regulátor polohy. Regulátory byly navrženy s ohledem na dobu vzorkování. Hodnoty regulátorů byly nejprve otestovány v nástroji MATLAB Simulink, který je schopen provádět simulace s nižším vzorkovacím časem než SIMIT. Změnou vzorkovacího kroku v Simulinku bylo následně ověřeno, zda je SIMIT schopen soustavu

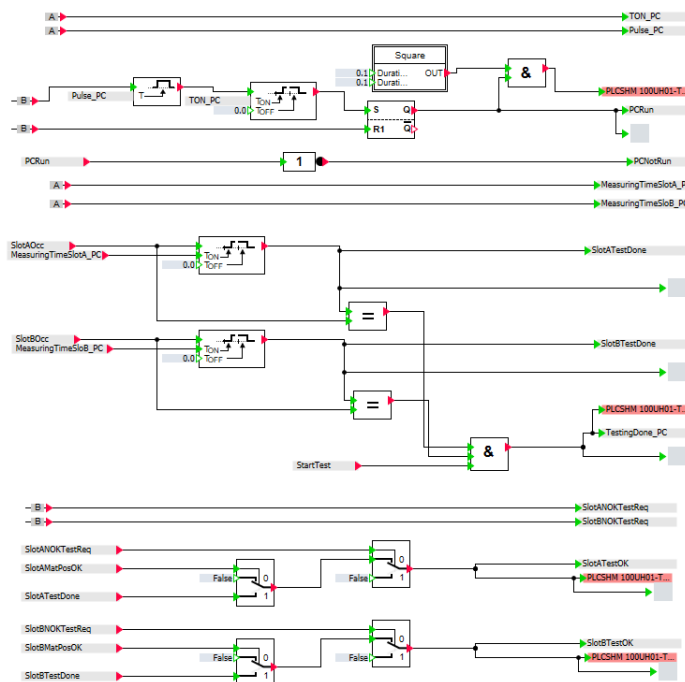
regulovat. Po úspěšné simulaci byly hodnoty nastaveny na vstup makra v nástroji SIMIT. Perioda vzorkování regulátoru byla nastavena na 10 ms pomocí *Time slice* 7. Výsledná regulace byla pro simulaci uspokojivá a průběhy byly uvedeny v příloze této práce.

Tabulka 7.3: *Parametry regulátorů CMMO jednotky*

Popis	Označení	$K_P$	$K_I$	$K_D$
Regulátor pozice	$G_{R1}$	3,172	0	0
Regulátor rychlosti	$G_{R2}$	$6,4625 \cdot 10^{-2}$	$1,9387 \cdot 10^{-1}$	0
Regulátor proudu	$G_{R3}$	$4,7029 \cdot 10^{-3}$	33,5719	0

## 7.4 Diagram měřicího PC

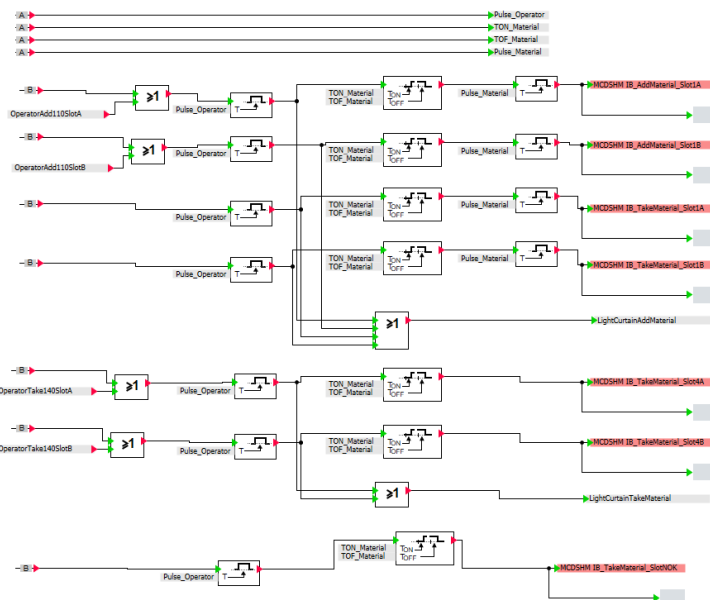
Chování měřicího PC bylo vytvořeno, na základě chování jednotky simulující měřicí PC s pomocí zpožďovačů, logických operátorů a bistabilních klopných obvodů. S náběžnou hranou signálu pro začátek měření jsou vyhodnoceny signály určující, na kterých pozicích je materiál. Následně po časové prodlevě dojde k odeslání signálu o výsledku měření pro první pozici a poté pro druhou pozici společně se signálem test dokončen. Po příchodu signálu značení senzorů dokončeno jsou uloženy stavy signálů určující výsledek označení pozice A a B. Po další časové prodlevě je zaslán signál zápis na pozici A dokončen. Následně s další časovou prodlevou je zaslán signál zápis na pozici B dokončen společně se signálem zápisy dokončeny. Pro simulaci NOK dílu se v diagramu nachází tlačítka pro vyhodnocení špatného měření nebo zápisu. NOK díl je také vyhodnocen v případě, že se díl nenachází na správné pozici kdykoliv během měření. Pozice je vyhodnocena ve fyzikálním a kinematickém modelu, viz. 6.3.



Obrázek 7.16: Část diagramu chování měřicího PC v SIMIT

## 7.5 Diagramy interakce

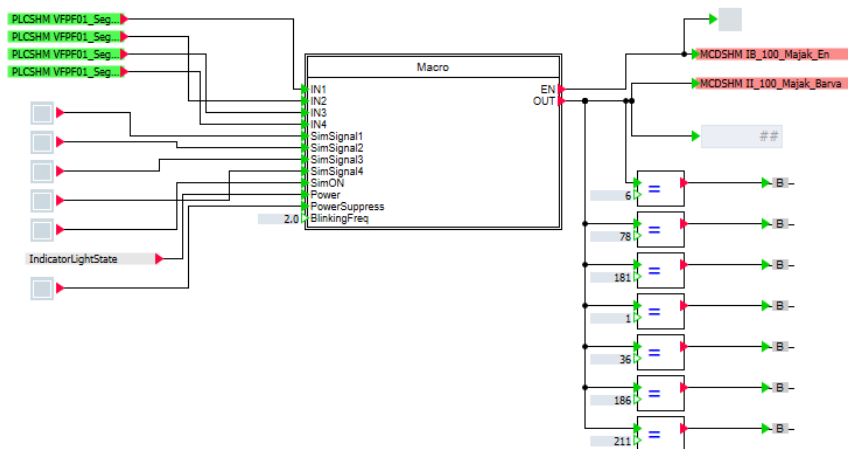
Mezi diagramy interakce patří všechny možné operace, které by bylo možné se strojem provádět jako ve skutečném světě. Byly zde vytvořeny diagramy pro operaci s materiálem, simulace otevření dveří, stisk tlačítek nouzového zastavení, protnutí bezpečnostní závory atd.



Obrázek 7.17: Diagram operací s materiálem v SIMIT

## 7.6 Diagramy vizualizačních prvků

Pro vizualizaci zařízení jako je maják nebo podsvícená tlačítka byly vytvořeny vizualizační diagramy. Pro potvrzovací tlačítko a maják byly vytvořeny makra. Makro pro maják obsahuje 3 vstupní signály, jejichž kombinace určuje, jaké číslo bude na výstupu. Výstup je napojen na signál v NX, který se postará o obarvení povrchu majáku. Čtvrtý binární vstup makra rozbliká maják frekvencí danou na příslušném vstupu *BlinkingFreq*. Barvy majáku byly nastaveny jako u skutečného zařízení podle dokumentace výrobce. Makro také obsahuje simulační signály umožňující snadné testování.



Obrázek 7.18: Diagram majáku v SIMIT

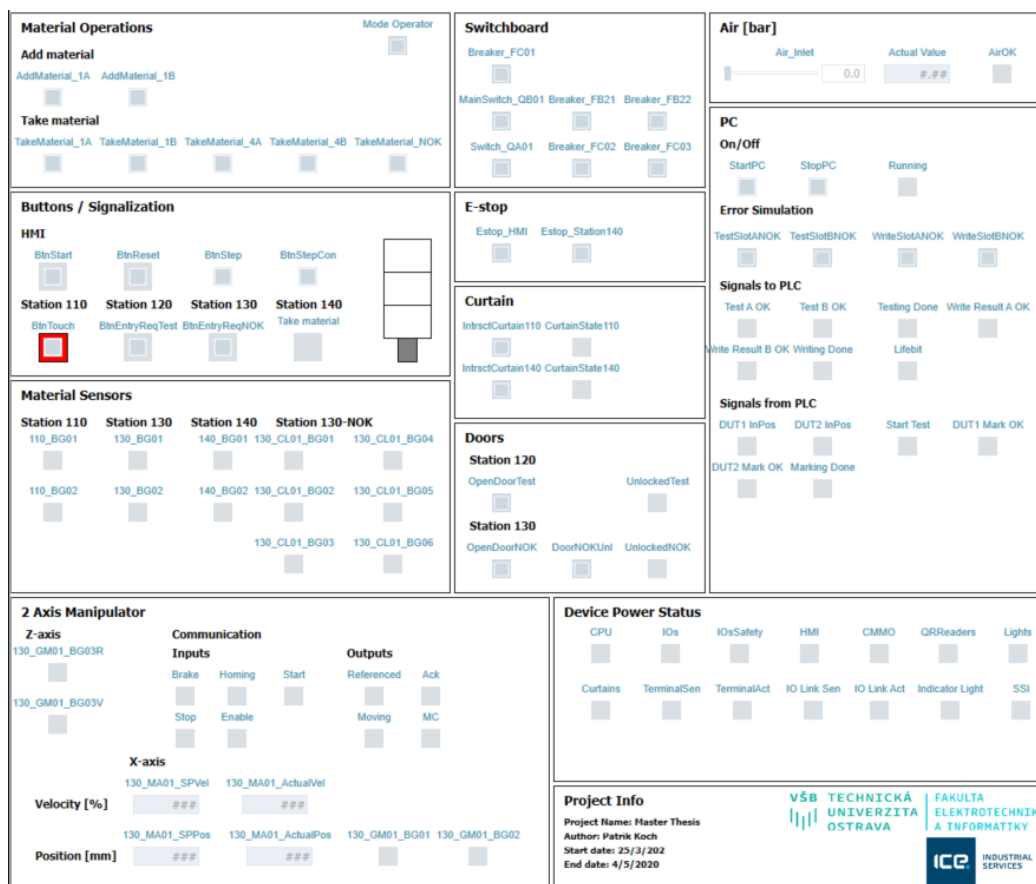


Tabulka 7.4: Ovládání osvětlení majáku

Bit 0-2	Bit 3	Barva
000	Aktivace blikání	Vypnuto
001		Zelená
010		Červená
011		Žlutá
100		Modrá
101		Oranžová
110		Definovaná uživatelem
111		Bílá

## 7.7 Hlavní přehledový diagram

Pro snadnou interakci se strojem byl vytvořen přehledový diagram. Pomocí diagramu lze snadno vložit nebo odebrat materiál, stisknout jakékoliv tlačítko, pozorovat stav majáku a indikátorů, sledovat stavy senzorů přítomnosti materiálu, sledovat stav manipulátoru, nahodit nebo shodit konkrétní jističe, stisknout tlačítko nouzového zastavení, záměrně protnout závoru, otevřít dveře, sledovat, zda je k zařízením přivedeno napájení, nastavit tlak na přívodu vzduchu nebo zapnout a sledovat výměnu signálů s měřícím PC.



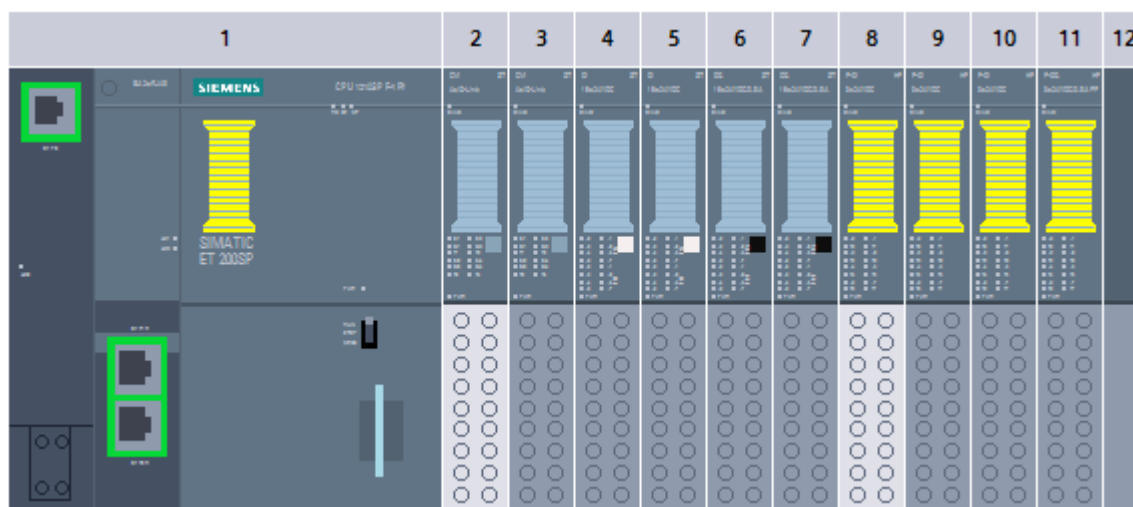
Obrázek 7.19: Hlavní přehledový diagram v SIMIT

## 8. Realizace řídicí aplikace

Řídicí aplikace byla vytvořena ve vývojovém prostředí TIA Portal V15.1 v programovacím jazyce kontaktních schémat. HW konfigurace a I/O tagy byly vytvořeny na základě elektro-dokumentace. Program byl vytvořen ve firemním standardu MICE. Sekvence byly vytvořeny pomocí tabulky pracovního cyklu stanice.

### 8.1 HW konfigurace

Ze začátku byla vytvořena hardware konfigurace odpovídající zvolenému PLC dle elektro-dokumentace. Pro aplikaci byl využit řídicí systém SIMATIC ET 200SP značky Siemens.



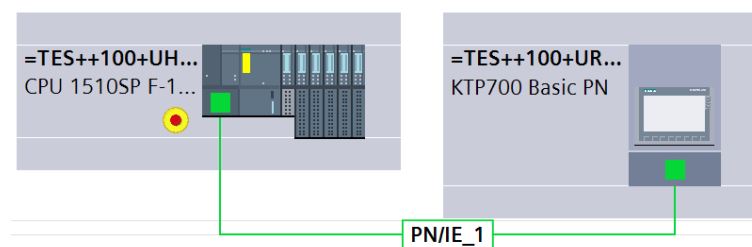
Obrázek 8.1: Hardware konfigurace PLC v TIA Portal

Po úspěšném vložení všech odpovídajících typů modulů, byl ve vlastnostech CPU povolen system and clock memory a zadáno Profinet jméno a IP adresa. Jednotlivým kartám byly přiřazeny I/O adresy a došlo k nastavení F-I/O modulů.

Tabulka 8.1: Hardware konfigurace PLC a adresy

Slot	Modul	Číslo výrobku	Firmware	Vstupní adresy	Výstupní adresy
1	CPU 1510SP F-1 PN	6ES7 510-1SJ01-0AB0	V2.5		
2	CM 4xIO-Link	6ES7 137-6BD00-0BA0	V2.2	100..131	100..131
3	CM 4xIO-Link	6ES7 137-6BD00-0BA0	V2.2	132..163	132..163
4	DI 16x24VDC ST	6ES7 131-6BH01-0BA0	V0.0	0..1	
5	DI 16x24VDC ST	6ES7 131-6BH01-0BA0	V0.0	2..3	
6	DQ 16x24VDC/0.5A ST	6ES7 132-6BH01-0BA0	V0.0		0..1
7	DQ 16x24VDC/0.5A ST	6ES7 132-6BH01-0BA0	V0.0		2..3
8	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0	V1.0	500..505	500..503
9	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0	V1.0	506..511	506..509
10	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0	V1.0	512..517	512..515

11	F-DQ 8x24VDC/0.5A PP HF	6ES7 136-6DC00-0CA0	V1.0	518..523	518..523
----	----------------------------	---------------------	------	----------	----------



Obrázek 8.2: Hardware konfigurace Profinet sítě v TIA Portal

Do systému bylo také přidáno HMI. Jedná se o tlačítkový panel s dotykovou obrazovkou KTP700 Basic. Na HMI bylo nastaveno Profinet jméno a IP adresa. Nakonec bylo HMI připojeno k PLC pomocí Profinet rozhraní.

Tabulka 8.2: Hardware konfigurace HMI

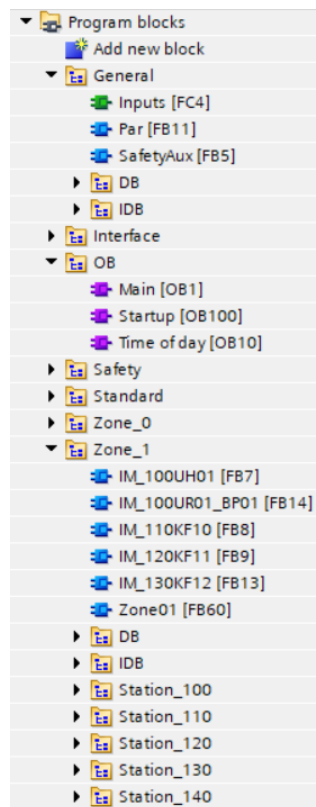
No.	Zařízení	Číslo výrobku	Firmware
1	HMI KTP700 Basic	6AV2 123-2GB03-0AX0	V15.0

Tabulka 8.3: IP adresy a Profinet jména síťových zařízení

No.	Zařízení	IP adresa	Profinet jméno
1	CPU 1510SP F-1 PN	172.19.7.1.	=tes++100+uh01-kf10
2	HMI KTP700 Basic	172.19.7.2.	=tes++100+ur01-kf01

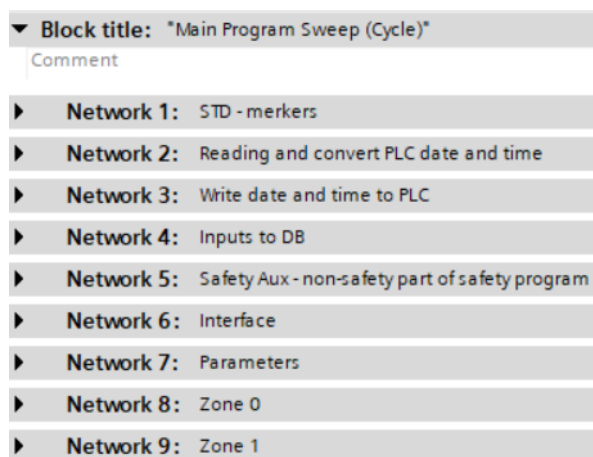
## 8.2 Struktura

V rámci programu byla vytvořena adresářová struktura. Využívá se možnosti vnořování bloků do složek pro požadované strukturování zlepšující orientaci v programu. Všechny použité organizační bloky jsou umístěny do složky „OB“. Ve složce „General“ se nachází obecné funkce programu. Komunikace s jinými zařízeními je umístěna ve složce „Interface“. Funkce týkající se bezpečnostního programu jsou umístěny ve složce „Safety“. „Zone0“ obsahuje FB rozvaděče, materiálůvých dat a počítání statistiky výroby. Ve složce „Zone1“ jsou umístěny FB pro čtení vstupů, hlavní FB a DBs zóny a následné strukturované stanice s jejich FB, DBs a případně sekvenčními FB.



Obrázek 8.3: *Struktura programu v TIA Portal*

Mezi hlavní části programové struktury patří systémová, diagnostická a parametrizační funkce. Jsou zde zpracovány vstupní signály, které jsou překlápěny do příslušných proměnných datových bloků. Je zde řešena komunikace s externím systémem. Následuje strukturované rozdělení zóna/stanice.

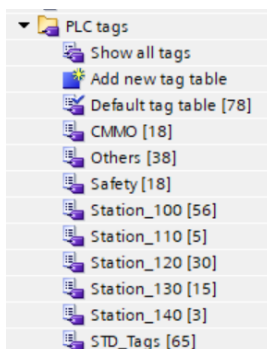


Obrázek 8.4: *Hlavní funkce programu OBI v TIA Portal*

### 8.3 Symbolika

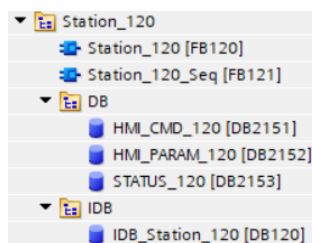
Symbolické proměnné byly vytvořeny jako tagy a proměnné v datových blocích. Mezi PLC tagy se nachází standardizované merkery, pro které je vymezená paměťová oblast 0-9 bytů. Většina standardizovaných merkerů je generována ve funkci „STD\_System“. Jedná se o Always True, Always

False, Commissioning a pulzní bity, které je možné využívat v programu. Mezi tagy se nachází také I/O signály. Všechny tagy byly rozděleny do tabulek podle stanic a typu pro lepší přehlednost.



Obrázek 8.5: *Vytvořené a importované PLC tagy v TIA Portal*

Pro uchovávání dat linky nebo stroje jsou využívány datové bloky s optimalizovaným přístupem. Pro každou zónu a stanici jsou vytvořeny tři datové bloky (statusové, povely vizualizace a parametry vizualizace). Statusový datový blok obsahuje stavy proměnných daného zařízení patřící do určité zóny nebo stanice. Jsou to např. stavy senzorů, motor v chodu nebo pneumatický válec vysunut. Povely vizualizace je příkazový datový blok, kde se nachází příkazy z vizualizace pro dané zařízení v rámci zóny nebo stanice. Jedná se např. zapnutí nebo vypnutí ručního režimu, pohyb motoru vpřed nebo změna polohy pneumatického ventilu. Datový blok pro parametry vizualizace obsahuje parametry pro dané zařízení, které jsou měnitelné z vizualizace např. dolní mez hladinového senzoru pro vyvolání poruchy nebo maximální otáčky motoru.

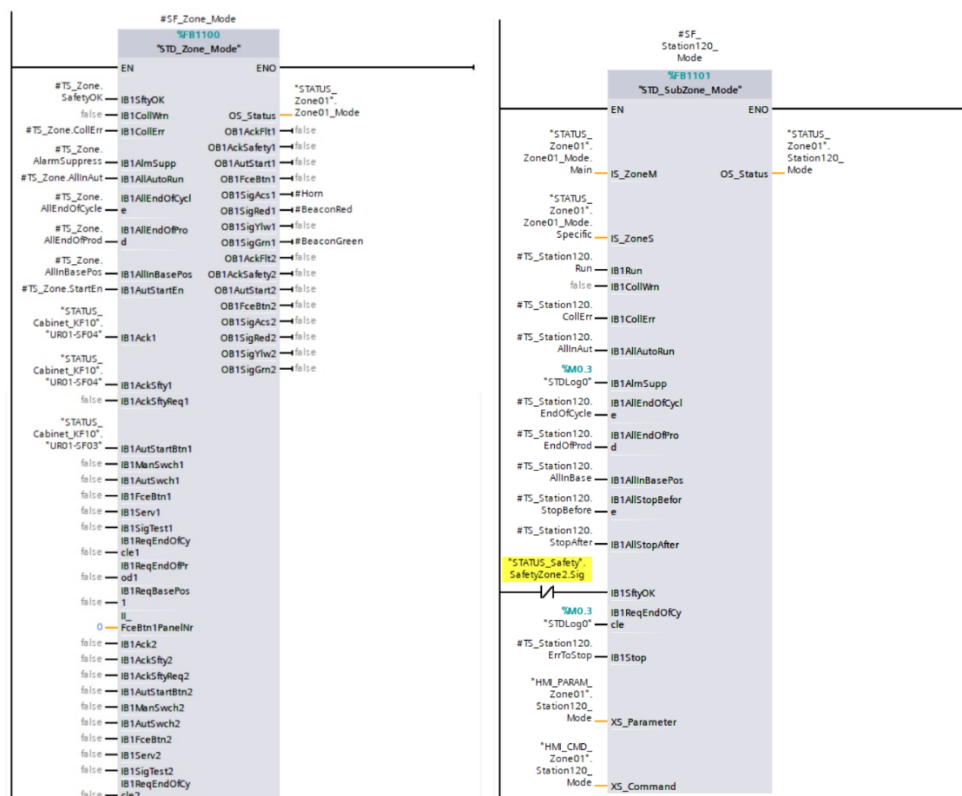


Obrázek 8.6: *Datové bloky stanice 120 v TIA Portal*

## 8.4 Zpracování vstupů

O načtení fyzických vstupů a uložení je do statusových DB se stará funkce „Inputs“. Oblast uvnitř funkce je dále strukturována podle zařízení (ET moduly, IO-Link master moduly). Každé zařízení je pak členěno do další úrovně zařízení (IO-Link slave moduly). Překlápění vstupních signálů probíhá pomocí standardizovaných funkcí.

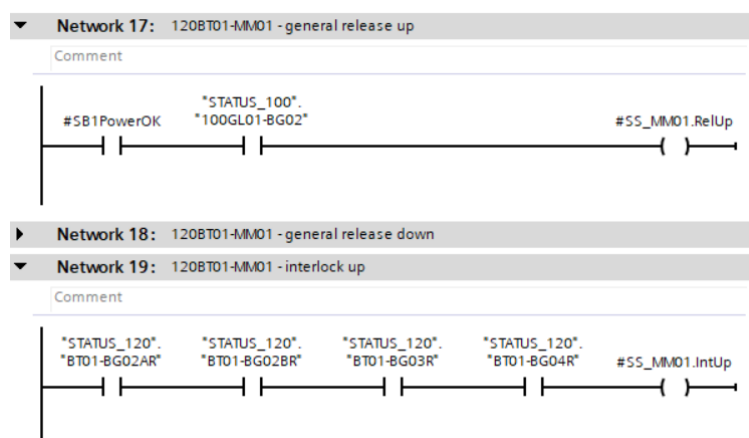




Obrázek 8.9: Řízení režimů zóny a stanice v TIA Portal

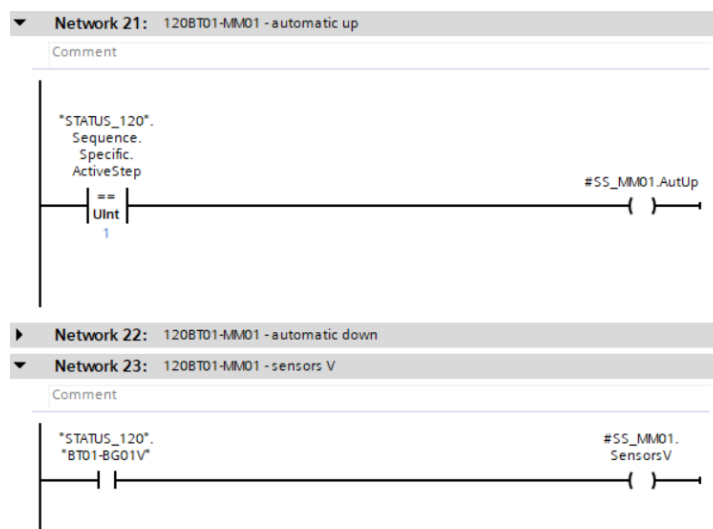
## 8.6 Akční členy

Pro řízení akčních členů jsou využity standardizované FB pro dané zařízení. V závislosti na umístění zařízení se FB zařízení nachází ve FB dané stanice. Pro změnu polohy ventilu je použit standardizovaný FB „STD\_Valve2WB“, pomocí kterého je po vyhodnocení vnitřní logiky aktivováno příslušné relé elektropneumatického ventilu. Pro řízení motoru je použit standardizovaný FB „STD\_Drive1“, který vyhodnocuje, zda jsou splněny patřičné podmínky pro aktivaci motoru. Výstup motoru je následně aktivován za pomoci FB „STD\_Motor“. Pro řízení lineárního pohonu manipulátoru byl využit FB poskytnutý výrobcem Festo.



Obrázek 8.10: Logika pro uvolnění a interlock pohybu ventilu v TIA Portal

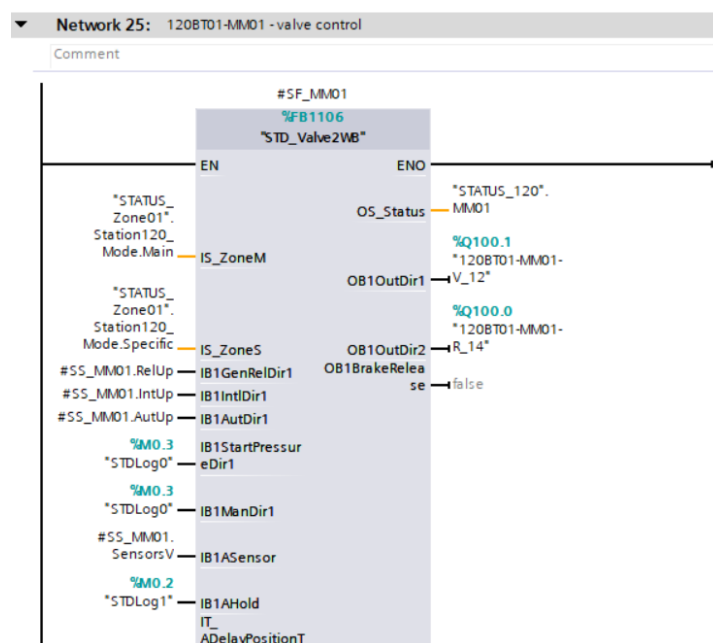
Pro umožnění změny polohy ventilu musí být aktivován příslušný signál uvolňující pohyb. Signál interlock blokuje změnu polohy pouze pokud se stroj nenachází v servisním režimu.



Obrázek 8.11: Logika pro automatický pohyb a zpracování senzoru v TIA Portal

Ventil je v automatickém režimu řízen za pomoci sekvencí. Otevření daného ventilu proběhne v kroku č. 1. Senzor vysunutého válce je napojen na příslušný signál senzoru.

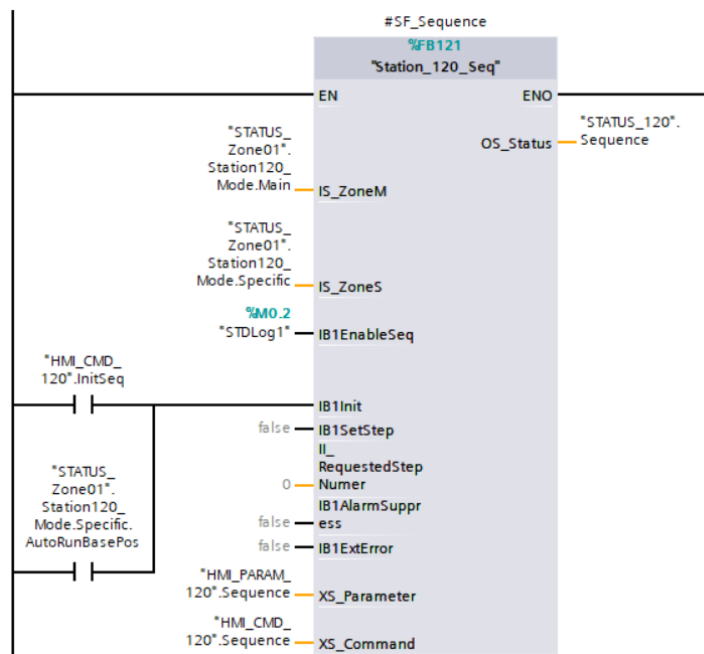
Ventil je řízen pomocí standardizovaného FB, na který jsou napojeny signály dané stanice, signály pro změnu polohy a časové konstanty do vyvolání poruchy v případě nestandardního chování.



Obrázek 8.12: Část FB pro řízení ventilu v TIA Portal

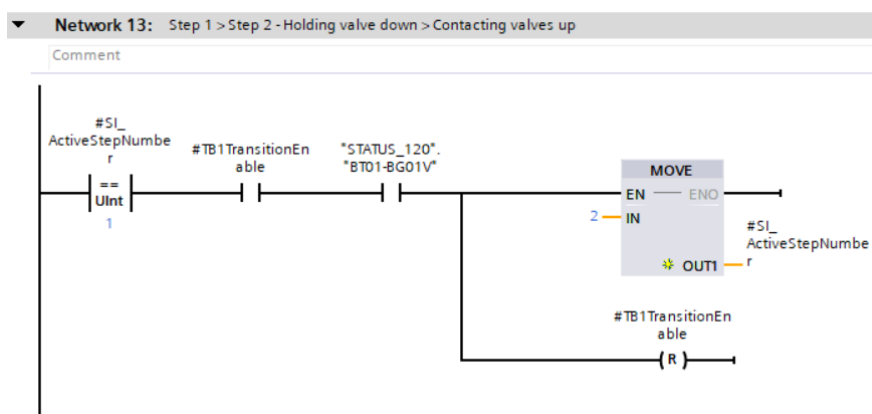


## 8.7 Sekvence



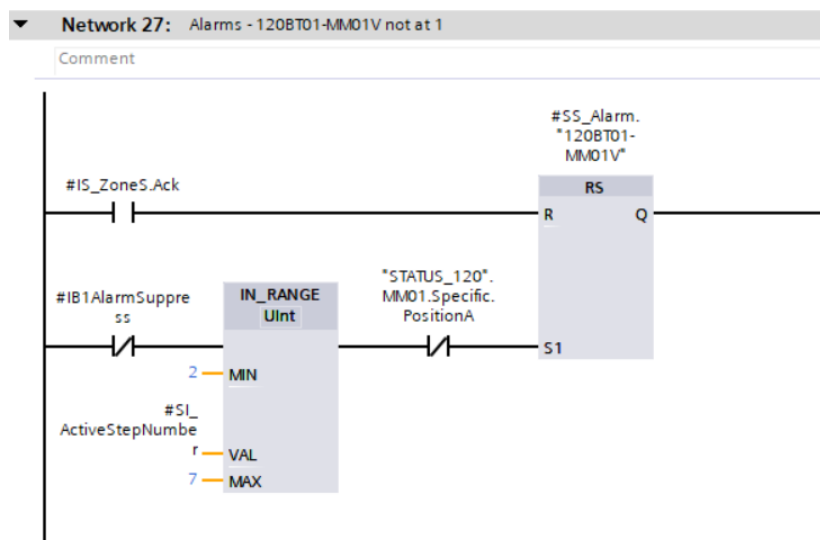
Obrázek 8.13: *Sekvenční FB v TIA Portal*

Pro řízení akčních členů stanic 120 a 130 bylo využito sekvenční řízení. Pro stanice byl vytvořen sekvenční FB. FB na začátku zpracovává signály pro povolení přesunu do následujícího kroku, zastavení před procesem, zastavení po procesu, konec cyklu a možnost nastavení kroku. Dále jsou řešeny kroky dané sekvence a následuje logika pro vyvolání sekvenčních alarmů a varování. Pro přechod do následujícího kroku musí být splněny podmínky přechodu a aktivní signál povolující přesun.



Obrázek 8.14: Podmínky přechodu do následujícího kroku v sekvenčním FB v TIA Portal

Sekvenční alarmy a varování jsou vyhodnocovány na základě kroku sekvence. Alarm upozorní operátora pomocí HMI a způsobí zastavení automatu, pokud je v chodu. Alarm je vyvolán, pokud se stanice nachází v konkrétním kroku a daný akční člen je mimo pozici. Sekvenční varování je vyvoláno jen v krocích, kdy dochází ke změně stavu daného akčního členu nebo v krocích, kdy se očekává nějaká akce.



Obrázek 8.15: Vyhodnocení sekvenčních alarmů v sekvenčním FB v TIA Portal

Odstartování jednotlivých sekvencí je vyvoláno změnou pozice otočného stolu. S náběžnou hranou na senzoru je detekována správná pozice karuselu a tím i odstartovány sekvence stanic 120 a 130. Karusel je automaticky otočen až po dokončení cyklu na stanicích 110, 120, 130 a 140.

### 8.7.1 Stanice 120

Tabulka 8.4: Výchozí stav stanice 120

Zařízení	Označení pohonu	Označení senzoru	Popis	Stav
Přidržení dílu A&B	120BT01-MM01	120BT01-BG01V	Vysunuto	0
		120BT01-BG01R	Zasunuto	1
Kontaktování dílu A&B	120BT01-MM02	120BT01-BG02AV	Vysunuto	0
		120BT01-BG02BV		0
		120BT01-BG02AR	Zasunuto	1
		120BT01-BG02BR		1
Značení OK dílu A	120BT01-MM03	120BT01-BG03V	Vysunuto	0
		120BT01-BG03R	Zasunuto	1
Značení OK dílu B	120BT01-MM04	120BT01-BG04V	Vysunuto	0
		120BT01-BG04R	Zasunuto	1

Pro vytvoření sekvenčního řízení stanice 120 byly vytvořeny tabulky popisující výchozí stav stanice a požadovaný pracovní cyklus stanice. V Tabulka 8.4 je uvedeno v jaké pozici jsou dané pohony a v jakém stavu jsou konkrétní senzory při začátku pracovního cyklu.

Tabulka 8.5 obsahuje přehledný popis všech kroků sekvence. Ve sloupci zařízení a akce je uvedeno jaký pohyb je v daném kroku proveden. Následující sloupec určuje podmínku přechodu, přičemž buňky v horizontálním směru jsou brány jako logický součin a buňky ve vertikálním směru jsou brány jako logický součet. Po splnění podmínky proběhne přechod do příslušného kroku uvedeného v posledním sloupci.

Tabulka 8.5: Pracovní cyklus stanice 120

Krok	Zařízení	Akce	Podmínka přechodu				Přechod do kroku
0	-	-	Start Cycle				1
1	Přidržení dílu A&B	Vysunout	120BT01-BG01V				2
2	Kontaktování dílu A&B	Vysunout	120BT01-BG02AV		120BT01-BG02BV		3
3	PC Měření	Start Test	Test Done		Slot A OK		4
					Slot B OK		
					Slot A NOK	Slot B NOK	7
4	Značení OK dílu A	Vysunout	120BT01-BG03V		Slot A OK	Slot B NOK	5
	Značení OK dílu B	Vysunout	120BT01-BG04V		Slot A NOK	Slot B OK	
			120BT01-BG03V	120BT01-BG04V	Slot A OK	Slot B OK	
5	Značení OK dílu A	Zasunout	120BT01-BG03R		120BT01-BG04R		6
	Značení OK dílu B	Zasunout					
6	PC Měření	Marking Done	Write Result Done				7
7	Kontaktování dílu A&B	Zasunout	120BT01-BG02AR		120BT01-BG02BR		8
8	Přidržení dílu A&B	Zasunout	120BT01-BG01R				9
9	-	Cycle Done	Cycle Done				0

Pomocí pracovního cyklu byly vytvořeny kroky v sekvenčním FB. Do sekvencí byl později zahrnut i přechod 0 > 9, jehož podmínkou je odstartování cyklu a zároveň prázdné sloty lůžka. Cyklus stanice by jinak nebyl nikdy dokončen. V případě, že by se ve slotech nenacházely díly, by totiž nedošlo k otočení karuselu.

► Network 10:	***** Sequence steps *****
► Network 11:	Step 0 > Step 1 - Init > Holding valve down
► Network 12:	Step 0 > Step 9 - Init > Cycle done
► Network 13:	Step 1 > Step 2 - Holding valve down > Contacting valves up
► Network 14:	Step 2 > Step 3 - Contacting valves up > PLC - PC mess Start
► Network 15:	Step 3 > Step 4 - PLC - PC mess Start > Mark slots - forward
► Network 16:	Step 3 > Step 7 - PLC - PC mess Start > Contacting valves down
► Network 17:	Step 4 > Step 5 - Mark slots - forward > Mark slots - backward
► Network 18:	Step 5 > Step 6 - Mark slots - backward > PLC - PC Marking done
► Network 19:	Step 6 > Step 7 - PLC - PC Marking done > Contacting valves down
► Network 20:	Step 7 > Step 8 - Contacting valves down > Holding valve up
► Network 21:	Step 8 > Step 9 - Holding valve up > Cycle done
► Network 22:	Step 9 > Step 0 - Cycle done > Init
► Network 23:	Step 10+ > Step 0 all not used steps to Init

Obrázek 8.16: Sekvence stanice 120 v TIA Portal

## 8.7.2 Stanice 130

Tabulka 8.6: *Výchozí stav stanice 130*

Zařízení	Označení pohonu	Označení senzoru	Popis	Stav
Z-osa manipulátor	130BT01-MM01	130GM01-BG03V	Vysunuto	0
		130GM01-BG03R	Zasunuto	1
Vakuum manipulátoru	130BT01-GS01	130GM01-BP01	Aktivováno	0
X-osa manipulátor	130XX01-TA01	130GM01-BG01	Koncová pozice	0
		130GM01-BG02	Koncová pozice	0
		Enkodér	Pozice	Home

Pro vytvoření sekvenčního řízení stanice 130 byly vytvořeny tabulky popisující výchozí stav stanice a požadovaný pracovní cyklus stanice. V Tabulka 8.6 je uvedeno v jaké pozici jsou dané pohony a v jakém stavu jsou konkrétní senzory při začátku pracovního cyklu.

Tabulka 8.7 obsahuje přehledný popis všech kroků sekvence. Ve sloupci zařízení a akce je uvedeno jaký pohyb je v daném kroku proveden. Následující sloupec určuje podmínku přechodu, přičemž buňky v horizontálním směru jsou brány jako logický součin a buňky ve vertikálním směru jsou brány jako logický součet. Po splnění podmínky proběhne přechod do příslušného kroku uvedeného v posledním sloupci.

Tabulka 8.7: *Pracovní cyklus stanice 130*

Krok	Zařízení	Akce	Podmínka přechodu		Přechod do kroku
0	-	-	Start Cycle	Slot A NOK	1
				Slot B NOK	
				Slot A OK   Slot B OK	21
1	-	-	Slot A NOK		2
			Slot A OK	Slot B NOK	11
2	X-osa manipulátor	Slot A	Pozice Slot A		3
3	Z-osa manipulátor	Vysunout	130GM01-BG03V		4
4	Vakuum manipulátoru	Aktivovat	130GM01-BP01		5
5	Z-osa manipulátor	Zasunout	130GM01-BG03R	Slot B OK	6
				Slot B NOK	7
6	-	Cycle Done	Cycle Done		7
7	X-osa manipulátor	Slot NOK	Pozice Slot NOK		8
8	Z-osa manipulátor	Vysunout	130GM01-BG03V		9
9	Vakuum manipulátoru	Deaktivovat	not 130GM01-BP01		10
10	Z-osa manipulátor	Zasunout	130GM01-BG03R	Slot B NOK	11
				Slot B OK	20
11	X-osa manipulátor	Slot B	Pozice Slot B		12
12	Z-osa manipulátor	Vysunout	130GM01-BG03V		13

13	Vakuum manipulátoru	Aktivovat	130GM01-BP01	14
14	Z-osa manipulátor	Zasunout	130GM01-BG03R	15
15	-	Cycle Done	Cycle Done	16
16	X-osa manipulátor	Slot NOK	Pozice Slot NOK	17
17	Z-osa manipulátor	Vysunout	130GM01-BG03V	18
18	Vakuum manipulátoru	Deaktivovat	not 130GM01-BP01	19
19	Z-osa manipulátor	Zasunout	130GM01-BG03R	20
20	X-osa manipulátor	Home	Pozice Home	0
21	-	Cycle Done	Cycle Done	0

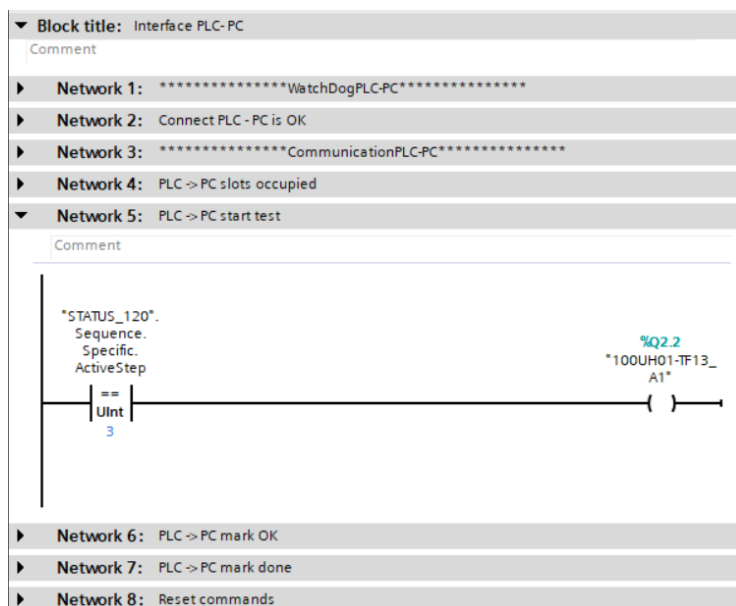
Pomocí pracovního cyklu byly vytvořeny kroky v sekvenčním FB. V sekvencích byl později upraven přechod 0 > 21, jehož podmínka nepočítala s prázdnými sloty lůžka. Cyklus stanice by jinak nebyl nikdy dokončen. V případě, že by se ve slotech nenacházely díly by totiž nedošlo k otočení karuselu.

▶	<b>Network 10:</b>	***** Sequence steps *****
▶	<b>Network 11:</b>	Step 0 > Step 1 - Init > Start the Sequence
▶	<b>Network 12:</b>	Step 0 > Step 21 - Init > Cycle done
▶	<b>Network 13:</b>	Step 1 > Step 2 - Start the Sequence > Axis X to Slot A
▶	<b>Network 14:</b>	Step 1 > Step 11 - Start the Sequence > Axis X to Slot B
▶	<b>Network 15:</b>	Step 2 > Step 3 - Axis X to Slot A > Axis Z down
▶	<b>Network 16:</b>	Step 3 > Step 4 - Axis Z down > Vacuum on
▶	<b>Network 17:</b>	Step 4 > Step 5 - Vacuum on > Axis Z up
▶	<b>Network 18:</b>	Step 5 > Step 6 - Axis Z up > Cycle done
▶	<b>Network 19:</b>	Step 5 > Step 7 - Axis Z up > Axis X to empty NOK slot
▶	<b>Network 20:</b>	Step 6 > Step 7 - Cycle done > Axis X to empty NOK slot
▶	<b>Network 21:</b>	Step 7 > Step 8 - Axis X to empty NOK slot > Axis Z down
▶	<b>Network 22:</b>	Step 8 > Step 9 - Axis Z down > Vacuum off
▶	<b>Network 23:</b>	Step 9 > Step 10 - Vacuum off > Axis Z up
▶	<b>Network 24:</b>	Step 10 > Step 11 - Axis Z up > Axis X to Slot B
▶	<b>Network 25:</b>	Step 10 > Step 20 - Axis Z up > Axis X to home
▶	<b>Network 26:</b>	Step 11 > Step 12 - Axis X to Slot B > Axis Z down
▶	<b>Network 27:</b>	Step 12 > Step 13 - Axis Z down > Vacuum on
▶	<b>Network 28:</b>	Step 13 > Step 14 - Vacuum on > Axis Z up
▶	<b>Network 29:</b>	Step 14 > Step 15 - Axis Z up > Cycle done
▶	<b>Network 30:</b>	Step 15 > Step 16 - Cycle done > Axis X to empty NOK slot
▶	<b>Network 31:</b>	Step 16 > Step 17 - Axis X to empty NOK slot > Axis Z down
▶	<b>Network 32:</b>	Step 17 > Step 18 - Axis Z down > Vacuum off
▶	<b>Network 33:</b>	Step 18 > Step 19 - Vacuum off > Axis Z up
▶	<b>Network 34:</b>	Step 19 > Step 20 - Axis Z up > Axis X to home
▶	<b>Network 35:</b>	Step 20 > Step 0 - Axis X to Home > Init
▶	<b>Network 36:</b>	Step 21 > Step 0 - Cycle done > Init
▶	<b>Network 37:</b>	Step 22+ > Step 0 all not used steps to Init

Obrázek 8.17: *Sekvence stanice 130 v TIA Portal*

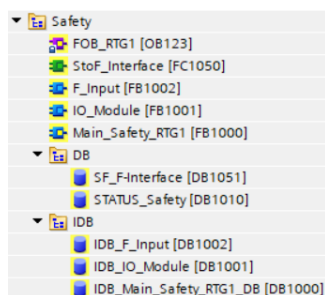
## 8.8 Obecné funkce

Mezi obecnými funkcemi se nachází FB pro materiálová data, FB pro komunikaci s externím měřicím PC a FB pro zpracování logiky mezi bezpečnostními a klasickými signály. Materiálová data jsou přenášena mezi pozicemi otočného stolu s náběžnou hranou senzoru pro detekci karuselu na pozici. Materiálová data obsahují informace jako slot aktivní, přítomnost dílu na pozici, díl označen, výsledek testování OK a výsledek zápisu OK. Komunikace s měřicím PC probíhá na základě kroku sekvence stanice 120. Přítomnost senzorů je určena z materiálových dat, jelikož se ve stanici nenachází senzory přítomnosti dílu. Obsahem FB pro zpracování logiky mezi bezpečnostními a klasickými signály je vyhodnocení kvitování a reintegrace pro daná zařízení, vyvolání alarmů z bezpečnostních signálů a logika bezpečnostního zámku.



Obrázek 8.18: Funkce pro komunikaci s měřicím PC v TIA Portal

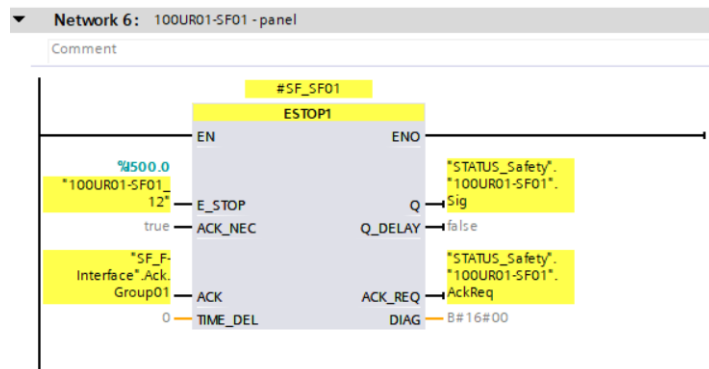
## 8.9 Bezpečnostní program



Obrázek 8.19: Struktura bezpečnostního programu v TIA Portal

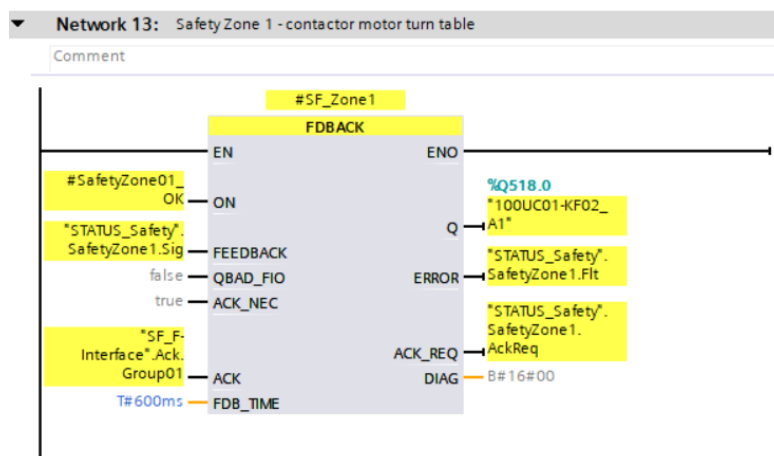
Jednotlivé části bezpečnostního programu se nachází ve složce „Safety“. Hlavní organizační blok cyklicky volá FB „Main\_Safety\_RTG1“. V tomto FB jsou volány FBs pro I/O moduly, pro bezpečnostní vstupy a pro bezpečnostní výstupy. Ve FB modulů je řešena reintegrace a zpracování stavu modulů. Ve funkci bezpečnostních vstupů jsou zpracovány vstupy jako e-stopky, bezpečnostní snímače,

světelné závory a zpětné vazby ze zařízení. Ve funkci bezpečnostních výstupů je na základě logiky aktivován bezpečnostní signál, který je vyhodnocován při sepnutí bezpečnostního výstupu v bezpečnostním programu. V hlavním FB je dále řešeno zpracování stavů stroje z hlediska bezpečnosti a vyhodnocení bezpečnostních zón.



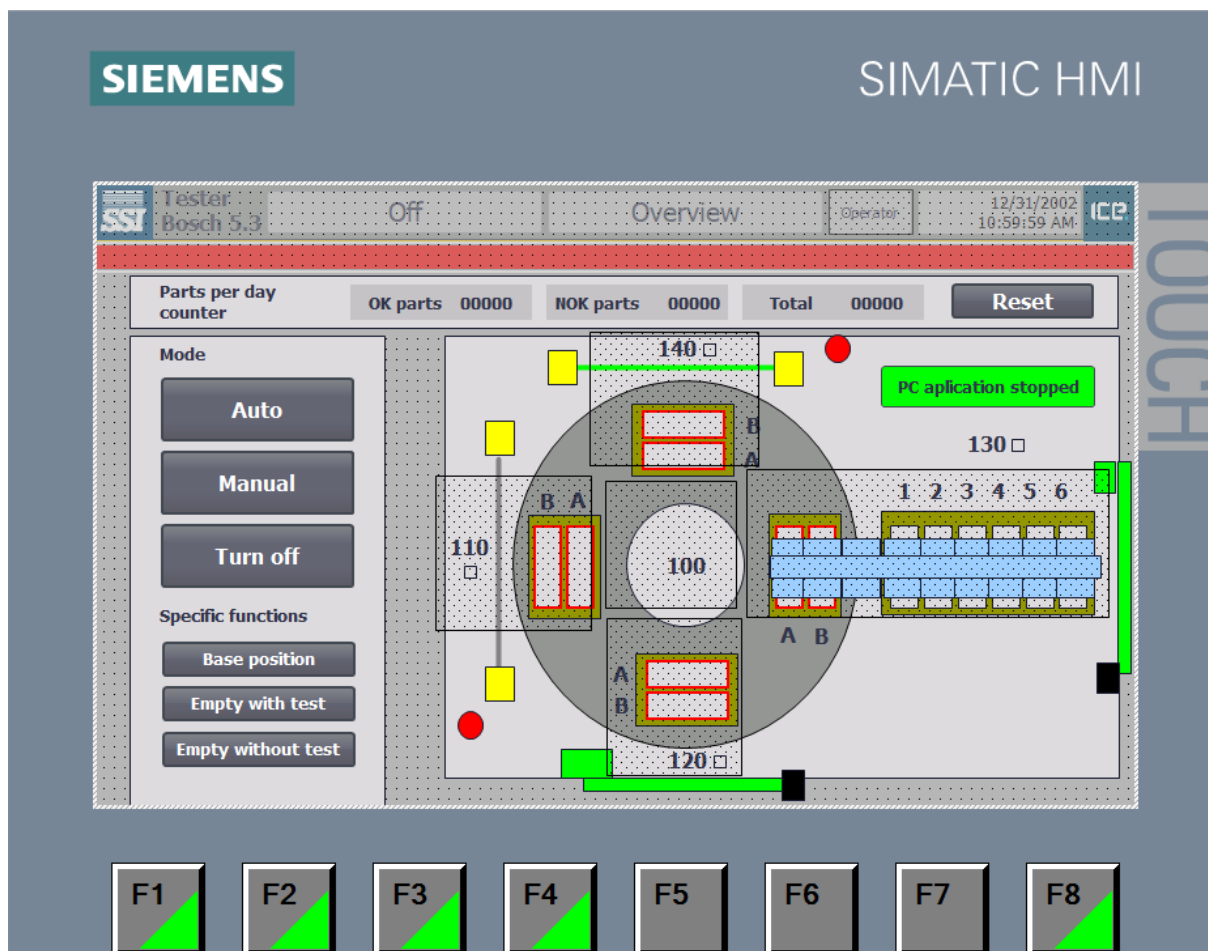
Obrázek 8.20: Zpracování signálu e-stopky v TIA Portal

Bezpečnostním program byl vytvořen s pomocí bezpečnostních instrukcí ESTOP1, FDBACK a ACK\_GL. Tyto instrukce jsou součástí *Step 7 Safety Advanced*. Heslo bezpečnostního programu bylo nastaveno jako „safety“.



Obrázek 8.21: Kontrola zpětné vazby při sepnutí stykače karuselu v TIA Portal

## 8.10 HMI



Obrázek 8.22: Přehledová obrazovka HMI v TIA Portal

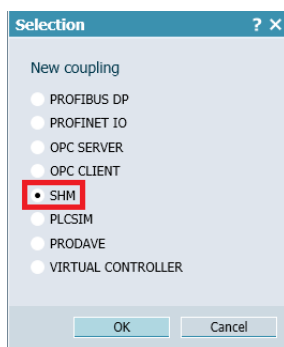
Za účelem vizualizace a řízení stroje pomocí HMI bylo vytvořeno několik obrazovek. Jedná se o přehledovou obrazovku, obrazovky stanic, obrazovku nastavení a obrazovku chybových hlášek. Přepínání mezi hlavními obrazovkami je uskutečněno pomocí fyzických tlačítek HMI. Pro zobrazení konkrétní obrazovky stanice je využito dotykového displeje na přehledové obrazovce. Všechny texty byly vytvořeny ve dvou jazycích. Přepínání mezi češtinou a angličtinou je provedeno pomocí fyzického tlačítka na HMI. Zapnutí servisního režimu a přístup na obrazovku nastavení a obrazovku stanice 120 a 130 je povolen pouze s patřičným oprávněním. Pro tato oprávnění byl vytvořen uživatel „admin“ s přístupovým heslem „admin“. HMI aplikace byla vytvořena v prostředí TIA Portal, tak jako PLC aplikace. Pro přístup k proměnným PLC bylo využito symbolické adresace. Chybové hlášky byly vygenerovány za pomoci MICE generátoru alarmů. Pro otestování HMI aplikace byl použit simulátor WinCC: HMI Runtime Simulation.



## 9. Vyhodnocení

V této závěrečné kapitole je popsáno nastavení výměny dat mezi jednotlivými nástroji, vytvořená aplikace pro zpracování dat z instance PLCSIM Advanced a výsledné virtuální provozní stroje.

### 9.1 Nastavení komunikace v SIMIT



Obrázek 9.1: Vytvoření propojení SHM v prostředí SIMIT

Pro vytvoření komunikace mezi SIMIT a NX MCD bylo zvoleno propojení (*Coupling*) SHM. Jedná se o komunikaci přes sdílenou paměť. Perioda aktualizace dat (*Time slice*) byla zvolena 8, což představuje čtení a zápis I/O každých 20 ms. Název SHM a mutexu byl zvolen viz. Obrázek 9.2. NX pro komunikaci pomocí SHM používá hlavičku obsahující popis signálů, proto bylo příslušné políčko zaškrtnuto. Pro rozložení bitu byl zvolen typ little endian.

MCDSHM	
Property	Value
Time slice	8
Shared memory name	SIMITMCDSHM
Mutex name	SIMITMCDSHMMutex
Signal description in header	<input checked="" type="checkbox"/>
Header size	1336
Big/Little Endian	little

Obrázek 9.2: Nastavení propojení SHM s NX v prostředí SIMIT

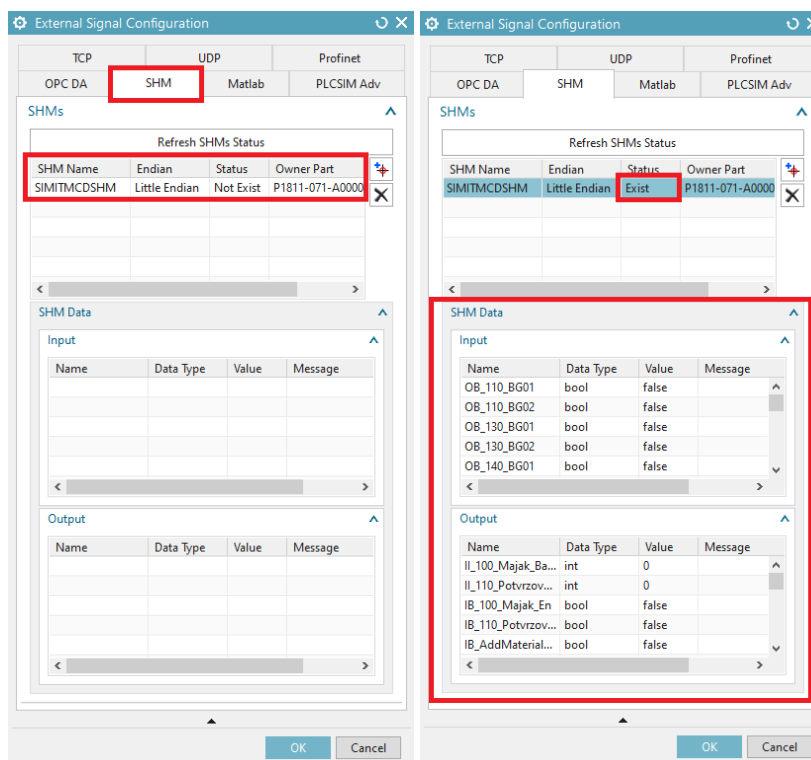
Pro vytvoření komunikace mezi SIMIT a C# aplikací bylo vytvořeno další propojení (*Coupling*) SHM. Perioda aktualizace dat (*Time slice*) byla zvolena 8, což představuje čtení a zápis I/O každých 20 ms. Název SHM a mutexu byl zvolen viz. Obrázek 9.3. Aplikace pro komunikaci pomocí SHM používá hlavičku o velikosti 8 bytu bez popisu signálů, příslušné políčko tedy nebylo zaškrtnuto. Pro rozložení bitu byl zvolen typ little endian, který je používán u SIMATIC PLC.

PLCSHM	
Property	Value
Time slice	8
Shared memory name	SIMITPLCSHM
Mutex name	SIMITPLCSHMMutex
Signal description in header	<input type="checkbox"/>
Header size	8
Big/Little Endian	little

Obrázek 9.3: Nastavení propojení SHM se C# aplikací v prostředí SIMIT

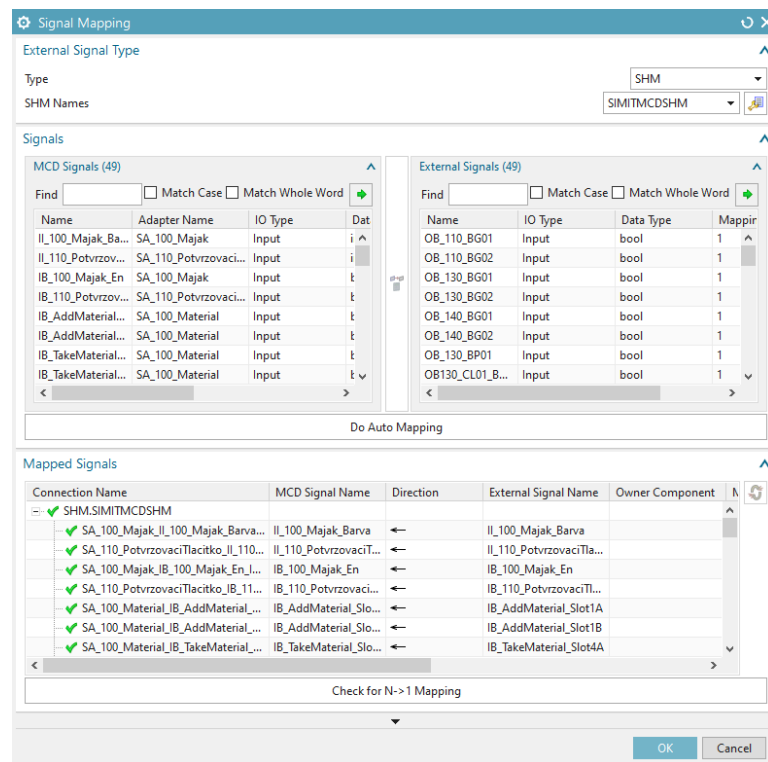
## 9.2 Nastavení komunikace v NX MCD

Pro nastavení propojení v NX MCD je potřeba otevřít *External Signal Configuration*. V záložce SHM bylo přidáno SHM propojení se stejným názvem jako v SIMIT (*SIMITMCDSHM*). Rozložení bitů bylo zvoleno také stejně jako v SIMIT little endian. Pokud v SIMIT není spuštěna simulace bude stav SHM označen jako neexistující. Po spuštění SIMIT simulace a obnovení stavu SHM v NX, je stav změněn na existující v případě provedení správného nastavení. Dostupná I/O data jsou poté zobrazena v okně níže, viz. Obrázek 9.4.



Obrázek 9.4: Externí konfigurace signálů v prostředí NX

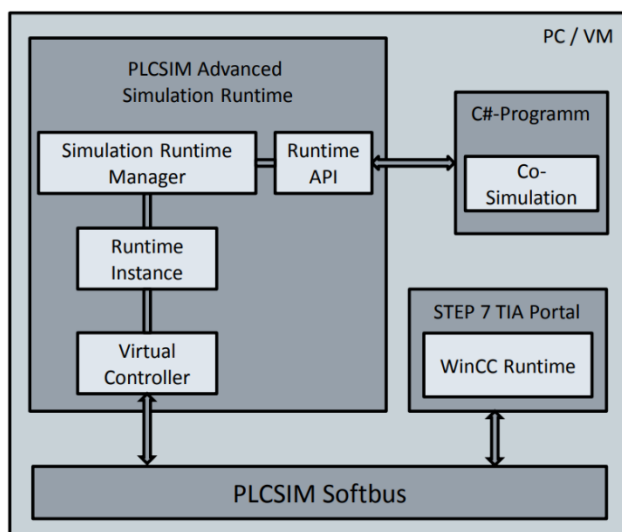
Pro napojení I/O signálů bylo otevřeno okno *Signal Mapping*, kde byla zvolena komunikace SHM a název vytvořeného SHM. Všechny signály mezi SIMIT a NX mohou být propojeny jeden po jednom manuálně nebo automaticky. Automaticky jsou propojeny pouze signály, které mají stejný název. Protože byly signály z NX do SIMIT exportovány bylo možné všechny signály propojit automaticky jedním klikem myši.



Obrázek 9.5: Napojení signálů v prostředí NX

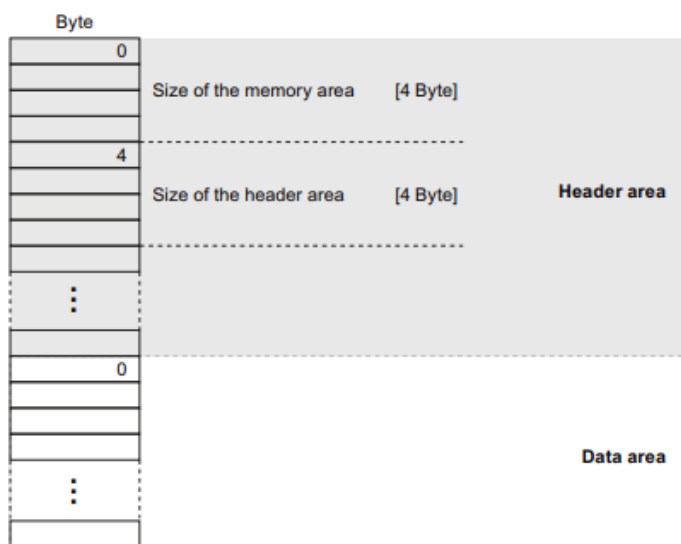
### 9.3 Vytvoření konzolové aplikace

Uživatelské rozhraní API umožňuje přistupovat do simulačního nástroje PLCSIM Advanced pomocí externí aplikace. Pomocí API lze vytvářet instance PLC, přepínat mezi jeho stavy a číst nebo zapisovat do proměnných skrz C# nebo C++ aplikaci. Pro přístup k proměnným je využito symbolických adres. Vytvořená aplikace byla vytvořena za účelem výměny dat mezi PLCSIM Advanced a SIMIT v programovacím jazyce C#.



Obrázek 9.6: Přehled funkčního principu PLCSIM Advanced [74]

Aplikace byla vytvořena v IDE Visual Studio 2019. Výstupy PLC jsou pomocí API čteny aplikací a zapisovány do sdílené paměti. Vstupy PLC jsou ze sdílené paměti přečteny a zapisovány pomocí API. Sdílená paměť má strukturu viz. Obrázek 9.7. Prvních 8 bytů je hlavičková oblast, kde první 4 byty určují celkovou velikost paměťové oblasti a následující 4 byty velikost hlavičkové oblasti. Celková velikost paměťového prostoru je 45 bytů a velikost hlavičky je 8 bytů. Pro datovou oblast byla vytvořena tabulka, kde byly jednotlivým signálům přiřazeny paměťové adresy. Tabulka je uvedena v příloze.



Obrázek 9.7: Struktura paměťové oblasti [35]

Na začátku programu se nachází globální proměnné určující velikost paměti, název instance PLC, periodu čtení/zápisu dat do sdílené paměti, vstupy a výstupy PLC atd.

```
const int headerTotalSizeOfMem = 45;    //Velikost paměti v bytech v hlavičce pro SIMIT
const int headerSizeOfHeader = 8;      //Velikost hlavičky v bytech v hlavičce pro SIMIT
const string PLC_instanceName = "PLC_1"; //Název instance PLC

static bool abortShm = false;    //Ukončení SHM
static bool comOK = false;    //Komunikace OK
static int comTime = 20;    //Perioda zápisu/čtení dat do/z SHM v ms
Počet odkazů: 3
public static MemoryMappedFile mmfG1 { get; set; } //SHM
Počet odkazů: 3
public static Mutex mutexG1 { get; set; } //SHM

static byte[] plcInputs = new byte[20]; //PLC vstupy
static byte[] plcOutputs = new byte[17]; //PLC výstupy

Počet odkazů: 99+
public static IInstance myInstance { get; set; } //PLCSIM Adv
static bool IsConfigured = false; //Tag table updated
```

Obrázek 9.8: Globální proměnné aplikace

Hlavní funkce *Main* vytvoří nové vlákno, které se postará o vytvoření, případně otevření paměťové oblasti a zavolá metodu starající se o připojení k instanci PLC. V případě úspěšné komunikace probíhá výměna dat. Ta je možná ukončit stiskem libovolné klávesy. Pokud se nepovede komunikaci zahájit zobrazí se příslušná hláška a aplikace je ukončena.

```

static void Main(string[] args)
{
    //Vytvoření/Otevření sdílené paměti
    Console.WriteLine("Creating / Opening a shared memory . . . ");
    Thread tShm = new Thread(() => Shm());
    tShm.Start();

    //Připojení k instanci PLCSIM API
    Console.WriteLine("Connecting to the PLCSIM instance . . . ");
    Plc(PLC_instanceName);

    if (myInstance != null)
    {
        //Možnost ukončení aplikace
        Console.WriteLine("");
        Console.WriteLine("Communication is running. ");
        Console.WriteLine("Press any key to quit . . . ");
        Console.WriteLine();
        Console.ReadKey(true || comOK);
        tShm.Interrupt();
        tShm.Join();
        Console.WriteLine();
        Console.WriteLine("Communication has been terminated. ");
    }
    else
    {
        tShm.Interrupt();
        tShm.Join();
        Console.WriteLine();
        Console.WriteLine("Communication start was not successful. ");
    }

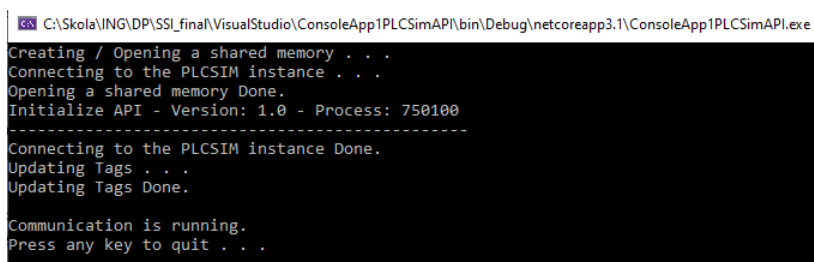
    abortShm = true;

    Console.WriteLine("Created by Patrik Koch. ");
    Console.WriteLine();
}

```

Obrázek 9.9: *Hlavní funkce aplikace Main*

V případě úspěšného vytvoření prostoru v paměti a napojení na instanci PLC běží konzolová aplikace viz. Obrázek 9.10. Úspěšné vytvoření nebo otevření prostoru ve sdílené paměti je oznámeno zprávou *Opening a shared memory Done*. V případě úspěšného připojení k instanci PLC nástroje PLCSIM Advanced je zobrazen řádek *Connecting to the PLCSIM instance Done*. Následně proběhne načtení tagů z PLC zakončeno zprávou *Updating Tags Done*. Pokud vše proběhne úspěšně je zobrazen řádek *Communication is running* a probíhá výměna dat. V případě, že během komunikace bude přehrán řídicí program, aplikace komunikaci pozastaví a provede aktualizaci tagů, dle nového řídicího programu. Poté je komunikace znovu obnovena.



```

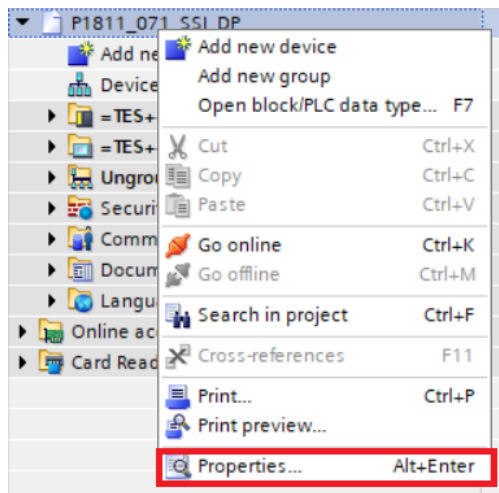
C:\Skola\ING\DP\SSI_final\VisualStudio\ConsoleApp1PLCSimAPI\bin\Debug\netcoreapp3.1\ConsoleApp1PLCSimAPI.exe
Creating / Opening a shared memory . . .
Connecting to the PLCSIM instance . . .
Opening a shared memory Done.
Initialize API - Version: 1.0 - Process: 750100
-----
Connecting to the PLCSIM instance Done.
Updating Tags . . .
Updating Tags Done.

Communication is running.
Press any key to quit . . .

```

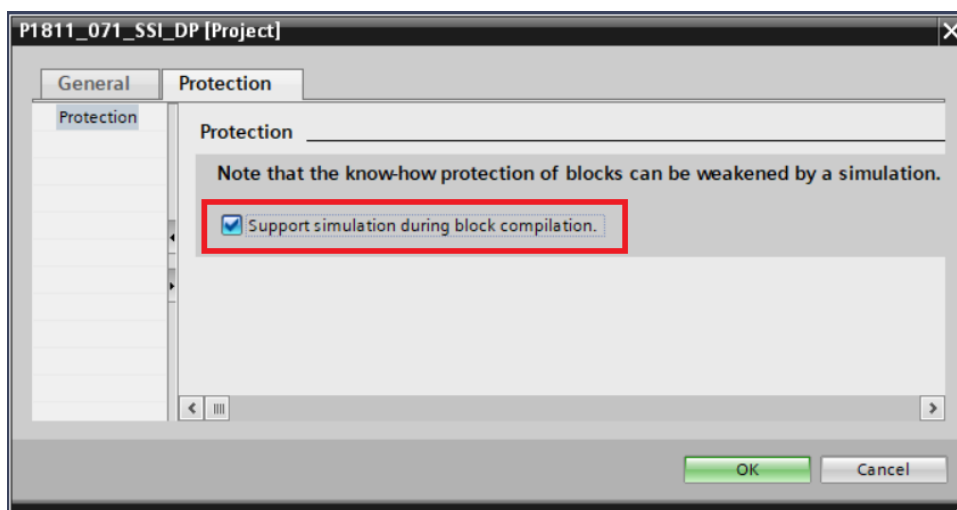
Obrázek 9.10: *Spuštěná aplikace s úspěšně vytvořenou komunikací*

## 9.4 Povolení simulace v TIA Portal



Obrázek 9.11: Otevření vlastností projektu TIA Portal

Pro simulaci řídicí aplikace vyvíjené v prostředí TIA Portal je potřeba zaškrtnout příslušný checkbox pro povolení simulace. Okno s povolením simulace se nachází ve vlastnostech projektu. V případě nezaškrtnutí políčka by nebylo dovoleno nahrát řídicí aplikaci do dané instance PLCSIM Advanced.

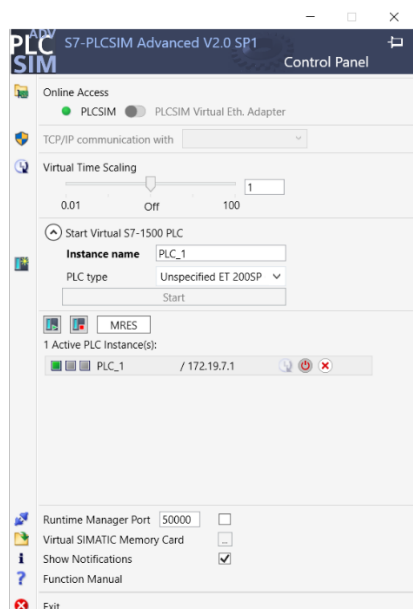


Obrázek 9.12: Povolení simulace v projektu TIA Portal

## 9.5 Virtuální zprovoznění

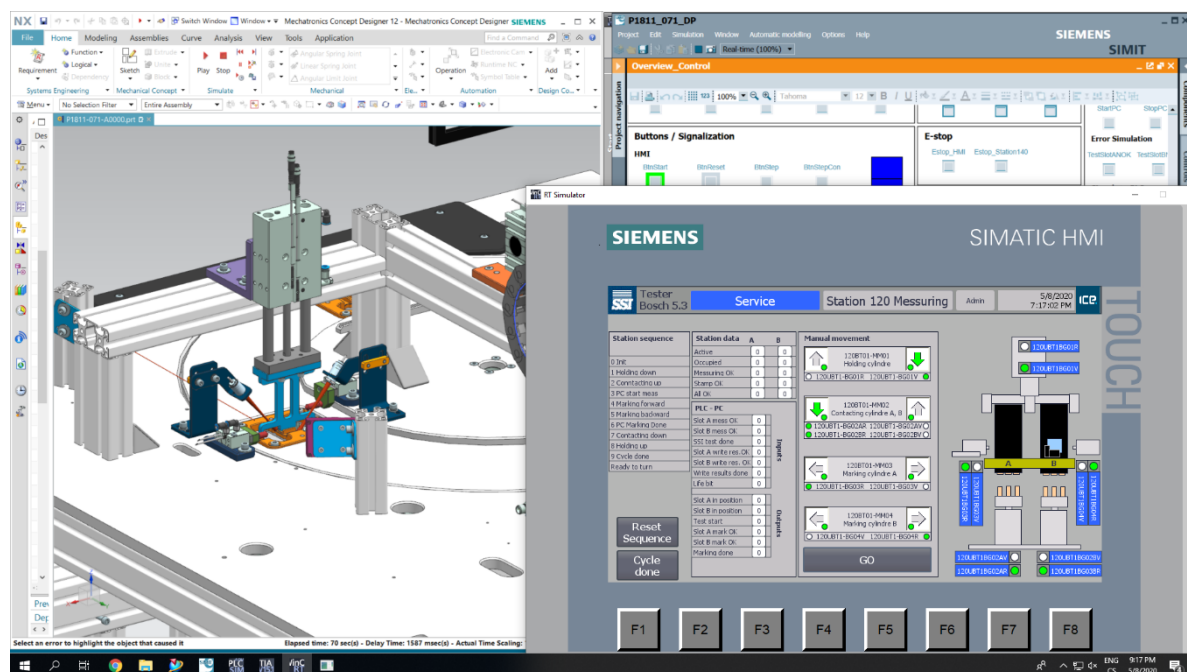
Virtuální zprovoznění stroje proběhlo na jednom PC simulační technikou SiL. Všechny nástroje běželi na PC s CPU AMD Ryzen 5 2600, 16 GB RAM 3200 MHz a grafickou kartou MSI GeForce GTX 760 OC 2 GB v prostředí OS Windows 10 Professional. Nejvíce docházelo k zatížení grafické karty, která běžela na plný výkon, hlavně kvůli detailního modelu v NX MCD. RAM počítače byla využívána okolo 68 % a CPU běžel v rozmezí 43 až 47 %.

Prvním krokem oživení modelu virtuálního stroje bylo spuštění nástroje PLCSIM Advanced 2.0 a vytvoření instance PLC řady ET200SP. Do vytvořené instance byla následně nahrána řídicí aplikace pomocí nástroje TIA Portal V15.1. Za účelem ověření HMI aplikace byla spuštěna jeho simulace v nástroji WinCC: HMI Runtime Simulation. Tento nástroj je součástí prostředí TIA Portal. V tomto bodě byl vytvořen automatizační model a bylo možné testovat komunikaci mezi PLC a HMI.



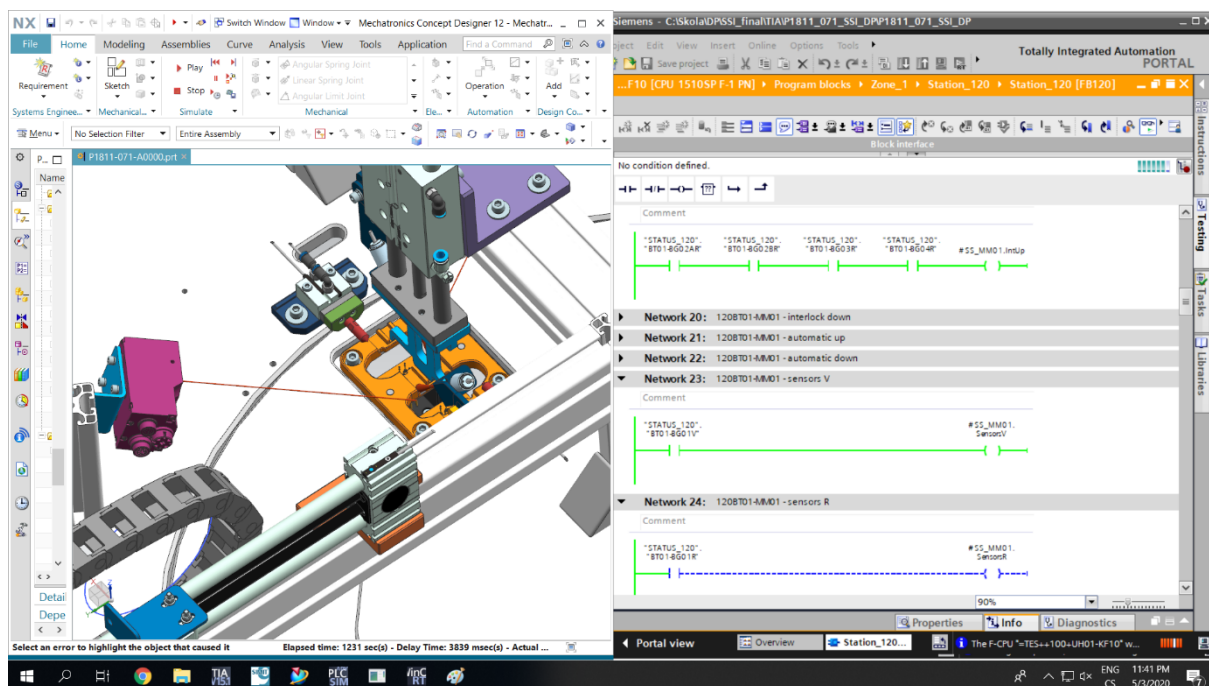
Obrázek 9.13: *Vytvoření instance PLCSIM Advanced*

Pro zprovoznění elektrického a behaviorálního modelu byla spuštěna simulace v prostředí SIMIT. Napojení na již vytvořenou instanci PLC je provedeno přes konzolovou aplikaci. Po spuštění této aplikace je zobrazen proces napojení k instanci PLC, načtení tagů a otevření sdíleného paměťového prostoru. Sdílený paměťový prostor je vytvořen konzolovou aplikací, v případě, že by došlo k spuštění simulace SIMIT až po konzolové aplikaci.



Obrázek 9.14: Řízení stroje pomocí simulovaného HMI

Po rozběhnutí komunikace mezi elektrickým a behaviorálním a automatizačním modelem bylo potřeba provést poslední krok. Spuštění fyzikálně-kinematického modelu je provedeno pomocí tlačítka pro spuštění simulace v prostředí NX MCD. Se spuštěnými simulačními nástroji je oživen virtuální stroj.

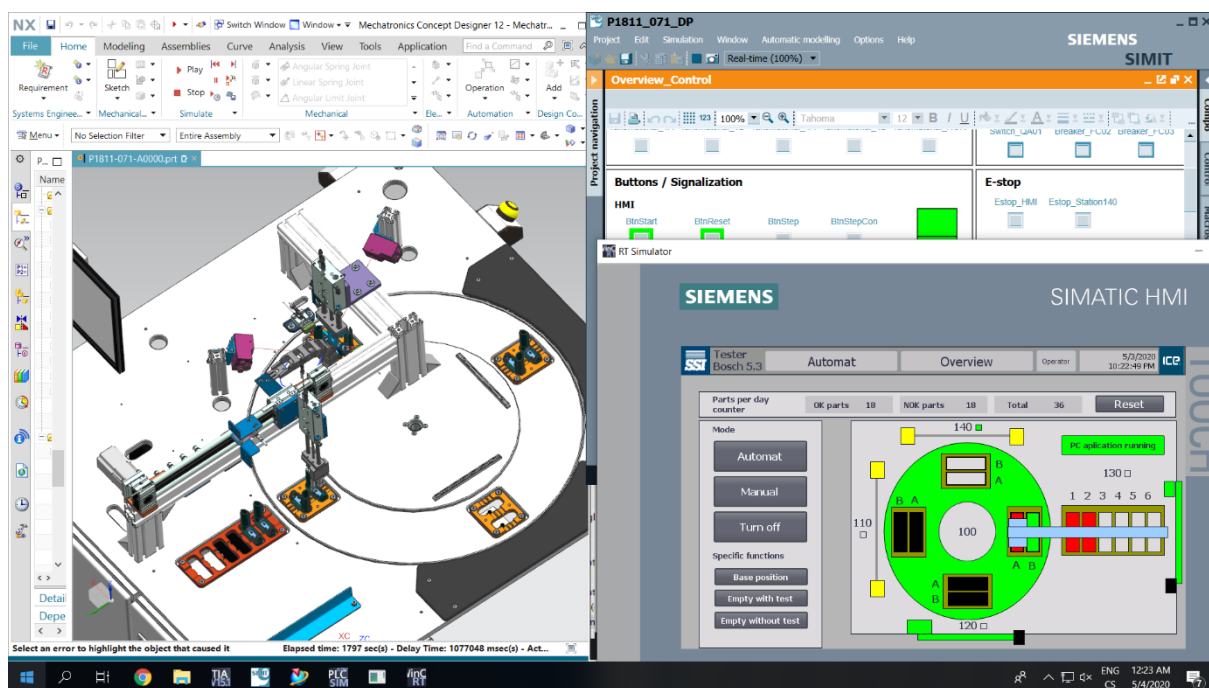


Obrázek 9.15: Monitorování řídicí aplikace během testů

Na stroji bylo provedeno virtuální zprovoznění podobně jako na fyzickém stroji. Nejprve byly nahozeny všechny jističe, byl přepnut hlavní přepínač, byl nastaven vstupní tlak a bylo zapnuto měřicí PC pomocí přehledového diagramu v SIMIT. Zprovoznění začalo kontrolou I/O signálů, kde byly



odhaleny drobné chyby v behaviorálním modelu. Signál senzoru vakua nebyl připojen na vstupní signál PLC, došlo ke špatnému vyhodnocení výstupního signálu *Moving* u CMMO jednotky a objevily se problémy s referencováním lineárního pohonu. Po těchto opravení chyb byl proveden test bezpečnostního programu, kde bylo zapotřebí nastavit vyšší časové konstanty u zpětnovazebních instrukcí, vzhledem k pomalejší komunikační výměně než se skutečným strojem. Následovalo zapnutí ručního režimu a nastavení časových konstant pro vyvolání chyby u FB akčních členů v řídicí aplikaci a naučení pozic lineárního pohonu v ose X. U ovládání motoru karuselu došlo k úpravě původní logiky vedoucí rychlejší reakci na senzor pro zastavení motoru, což bylo odhaleno vykreslením grafu v nástroji SIMIT. Pro ověření sekvencí byl spuštěn krokový automat, kde byly odhaleny a následně upraveny chybné části kódu způsobené vlivem kopírování. Bylo ověřeno vyvolání běžných a sekvenčních chyb a jejich hlášení na HMI. V NX lze do systému různě zasahovat a sledovat tak jeho chování, např. působit určitou silou proti pohybu válce. Nakonec byl stroj uveden do automatického režimu a byly testovány situace s neaktivním slotem na karuselu, deaktivace NOK slotů a speciální funkce vyprázdnění s testem a vyprázdnění bez testu. Výsledkem tohoto oživení je ověřená funkční řídicí aplikace.



Obrázek 9.16: *Virtuálně zprovozněný stroj v automatickém režimu*

Cílem této práce bylo také zjistit časovou náročnost při vytváření fyzikálně-kinematického a elektrického a behaviorálního modelu. Tato náročnost byla zaznamenána pro jednotlivé operace při vytváření patřičného modelu. Nejvíce času bylo stráveno u behaviorálního modelu při vytváření modelu chování pohonů. Zde bylo dopodrobna řešeno chování motoru karuselu a CMMO jednotky. Celkem bylo pro vytvoření obou simulačních modelu spotřebováno přibližně 109 hodin.

Při vytváření simulačních modelů jsem se s nástroji seznamoval, a proto si odhadnu říci, že tento čas by byl 2x nižší při vytváření těchto simulačních modelů podruhé. Čas by byl také snížen v případě, použití již připravených maker chování pro jednotlivá zařízení. Tyto makra by následně mohly být generovány do nástroje SIMIT již během vytváření fyzikálně-kinematického modelu.

Tabulka 9.1: Časová náročnost při vytváření simulačních modelů

Model	Operace	Čas [hod]
Fyzikálně-kinematický model	Fyzikální vlastností	3
	Pohony a senzory	8
	Operace s materiálem	16
	Signály	4
	Time-based simulace	3
	Ostatní (vizualizace, export, testy)	12
	<b>Celkem</b>	<b>46</b>
Elektrický a behaviorální model	Napájení a prvky rozvaděče	8
	Senzory a válce	4
	Pohony	31
	Měřicí PC	8
	Interakce	3
	Vizualizační prvky	3
	Hlavní přehledový diagram	6
	<b>Celkem</b>	<b>63</b>
<b>Celkový čas</b>		<b>109</b>

## Závěr

Cílem této práce bylo získat přehled o možnostech virtuálního zprovoznění, provést virtuální zprovoznění jednoúčelového stroje pro testování senzorů a zjistit časovou náročnost pro vytvoření fyzikálně-kinematického a elektrického a behaviorálního modelu. Byly získány poznatky z oblasti virtuálního zprovoznění a jeho různé možnosti byly popsány a analyzovány v teoretické části této práce. Jednoúčelový stroj byl úspěšně zprovozněn na jediném PC bez potřeby dodatečného HW a řízen řídicí aplikací pomocí simulovaného PLC a HMI. Z důvodu simulace PLC se jedná o použití simulační techniky Software-in-the-Loop. Na virtuálním modelu stroje byla aplikace otestována a optimalizována, což vedlo k odstranění nežádoucích chyb, ověření chování stroje během různých scénářů a ke vzniku odladěného a přehledného kódu. Pro vytvoření fyzikálně-kinematického a elektrického a behaviorálního modelu bylo potřeba přibližně 109 hodin. Během vývoje byl u jednotlivých kroků zapisován strávený čas z důvodu přehledného vyobrazení časové náročnosti jednotlivých fází vývoje.

Čas strávený na vývoji fyzikálně-kinematického modelu by mohl být redukován, pokud bych s nástrojem NX MCD měl dřívější zkušenost. Části fyzikálních vlastností nebyly zpočátku aplikovány správně, což vedlo k pozdějším nuceným úpravám. Nalezení funkčního způsobu upevnění materiálu vakuovým ejektorem zabralo přibližně 10 hodin, jelikož tato informace nebyla uvedena v manuálu. Během vývoje elektrického a behaviorálního modelu jsem měl s nástrojem SIMIT základní zkušenosti již ze své bakalářské práce, nicméně při opakovaném vytváření modelu by nebylo potřebné vytvářet nová makra, což by vedlo ke snížení času stráveném na vývoji simulačního modelu. Odhaduji si říci, že při vytváření obou simulačních modelů podruhé by bylo dosaženo 2x kratší doby.

Původně bylo počítáno s komunikací mezi nástroji SIMIT a PLCSIM Advanced pomocí OPC. Nicméně komunikaci se nepovedlo zprovoznit a z tohoto důvodu byla vytvořena konzolová aplikace. Použití aplikace vedlo k rychlejší komunikaci, jelikož aktualizace proměnných probíhá každých 20 ms, což je daleko rychlejší než pomocí rozhraní OPC. Hotový virtuální model stroje může být dále využit za účelem vzdělávání v rámci ICE Industrial Academy nebo pro vytvoření digitálního dvojčete po napojení fyzických signálů skutečného stroje.

Z praktického hlediska se podle mě jedná o budoucnost průmyslové automatizace a při použití patřičných generátorů elektrického a behaviorálního modelu během vytváření konstrukčního nebo fyzikálně-kinematického modelu se investovaný čas a finance do SW nástrojů vyplatí. Nicméně jinak to může být v případě, kdy je nutné vytvářet složité modely chování bez možnosti opakovatelného použití, zde je nutno zvážit, zda vytvoření simulačních modelů bude přínosem nebo spíše zbytečná zátěž. U prvního projektu pravděpodobně enormně naroste počet strávených hodin, avšak dojde k vytvoření několika diagramů chování zařízení pomocí maker, které u dalšího projektu nebude potřeba vytvářet znova, v případě, že budou použita stejná nebo podobná zařízení. V následujících větách jsou uvedeny výhody virtuálního zprovoznění. Řídicí aplikaci je možné otestovat i v případě, že stroj ještě není fyzicky smontován nebo zaslán do výroby. Je zkrácena doba zprovoznění fyzického stroje. Nedochází k úpravám řídicí aplikace až na stavbě nebo během krizových situací, čímž vznikne přehlednější a čitelnější kód. Je možné testovat různé inovace v řešení bez strachu z poškození stroje. Část doby programátora na stavbě je přesunuta do klidného prostředí kanceláře. Zákazníkovi je možné prezentovat virtuální chování stroje a včas identifikovat případné nejasnosti. Jako nevýhodu virtuálního zprovoznění lze považovat vysokou počáteční investici, musí být detailně vyjasněno chování jednotlivých částí systému, bez konstrukčních dat není možné stroj virtuálně oživit a u komplexních simulací je potřeba vytvářet složité modely chování.

Dalším přínosem této diplomové práce by mohlo být využití poznatků pro studijní účely. Virtuální zprovoznění je komplexní záležitost a společně s nástrojem NX MCD, který je zároveň konstrukčním nástrojem a umožňuje rychlý a snadný vývoj konceptu stroje, je ideálním kandidátem pro zadání univerzitního týmového projektu. Studenti by mohli podle prostého zadání vytvořit koncept stroje, ke kterému by byl následně vytvořen elektrický a behaviorální model společně s řídicí a vizualizační aplikací. Je možné také vytvořit virtuální modely strojů v laboratoři programovatelných automatů, které by následně mohly být využity studenty k testování jejich řídicích aplikací.

## Literatura

- [1] BANDYOPADHYAY, Susmita a BHATTACHARYA, Ranjan. *Discrete and Continuous Simulation: Theory and Practise*. Boca Raton, FL : CRC Press, 2014. ISBN: 978-1-4665-9639-9.
- [2] BANKS, Jerry, a další. *Discrete-Event System Simulation*. Upper Saddle River, NJ : Prentice Hall, 2004. ISBN: 0131446797.
- [3] SMITH, Roger D. Simulation: The Engine Behind the Virtual World. *Model Benders*. [Online] Prosinec 1999. [Citace: 25. Listopad 2016.] <http://www.modelbenders.com/papers/sim2000/SimulationEngine.PDF>.
- [4] Gomes, Cláudio. *Foundations for Co-simulation*. 2015.
- [5] SCHLAGER, Martin. *Hardware-in-the-Loop Simulation : A Scalable, Component-based, Time-triggered Hardware-in-the-loop Simulation Framework*. Saarbrücken : VDM Verlag Dr. Müller GmbH & Co. KG, 2011. ISBN: 978-3-8364-6216-7.
- [6] Using V Models for Testing. *SEI Insights*. [Online] 11. Listopad 2013. [Citace: 19. Prosinec 2016.] [https://insights.sei.cmu.edu/sei\\_blog/2013/11/using-v-models-for-testing.html](https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html).
- [7] Model-Based Systems Engineering and Control System Development via Virtual Hardware-in-the-Loop Simulation. *Autonomie*. [Online] Říjen 2010. [Citace: 20. Prosinec 2016.] [http://www.autonomie.net/docs/5%20-%20Presentations/CIL/modelbased\\_systems\\_engineering\\_ppt.pdf](http://www.autonomie.net/docs/5%20-%20Presentations/CIL/modelbased_systems_engineering_ppt.pdf).
- [8] MIL simulation - Model in the loop. *Rexcontrols*. [Online] [Citace: 19. Prosinec 2016.] <https://www.rexcontrols.com/mil-simulation>.
- [9] SIL simulation - Software in the loop. *Rexcontrols*. [Online] [Citace: 19. Prosinec 2016.] <https://www.rexcontrols.com/sil-simulation>.
- [10] Model-based Design Definition of Terms. *Autonomie*. [Online] [Citace: 19. Prosinec 2016.] [http://www.autonomie.net/%5C/references/model\\_based\\_design\\_defs\\_24c.html](http://www.autonomie.net/%5C/references/model_based_design_defs_24c.html).
- [11] PIL simulation - Processor in the loop. *Rexcontrols*. [Online] [Citace: 19. Prosinec 2016.] <https://www.rexcontrols.com/pil-simulation>.
- [12] HIL simulation - Hardware in the loop. *Rexcontrols*. [Online] [Citace: 19. Prosinec 2016.] <https://www.rexcontrols.com/hil-simulation>.
- [13] Systems, GSE. GSE Systems. *What is Virtual Commissioning and why is the so much buzz about it...* [Online] 24. Srpen 2017. [Citace: 15. Listopad 2019.] <http://www.gses.com/blog/simulation-training/virtual-commissioning/>.
- [14] Bernard, Dave. ERP information, news and tips - SearchERP. *What is virtual commissioning? Definition from WhatIs.com*. [Online] [Citace: 15. Listopad 2019.] <http://www.searcherp.techtarget.com/definition/virtual-commissioning>.

- [15] WinMOD. *WinMOD Summary 2019: Virtual Commissioning and more with the Digital Twin*. [Dokument] Hennigsdorf : autor neznámý, 2019.
- [16] Visual Components. Increasing control software quality with virtual commissioning. *Visual Components - A world leader in 3D automation*. [Online] 18. Březen 2016. [Citace: 15. Listopad 2019.] <http://www.visualcomponents.com/insights/articles/increasing-control-software-quality-with-virtual-commissioning/>.
- [17] SIEMENS. Virtual commissioning | Digital Enterprise for discrete industries. *Siemens*. [Online] [Citace: 15. Listopad 2019.] <http://www.new.siemens.com/global/en/company/topic-areas/digital-enterprise/discrete-industry/virtual-commissioning.html>.
- [18] Haltmar, Jaroslav, Baumruk, Martin a Kváč, Josef. *Robotika a Virtuální zprovoznění*. [Webinář] Praha : Siemens, 7. Prosinec 2018.
- [19] Tao, Fei, Zhang, Meng a Nee, A.Y.C. *Digital Twin Driven Smart Manufacturing*. místo neznámé : Matthew Deans, 2019. ISBN: 978-0-12-817630-6.
- [20] Mikell, Matthew a Clark, Jen. IBM. *Cheat sheet: What is Digital Twin?* [Online] 4. Leden 2018. [Citace: 12. Listopad 2019.] <http://www.ibm.com/blogs/internet-of-things/iot-cheat-sheet-digital-twin/>.
- [21] SIEMENS. Siemens Digital Industries Software. *Digital Twin | Siemens*. [Online] Siemens. [Citace: 12. Listopad 2019.] <http://www.plm.automation.siemens.com/global/en/our-story/glossary/digital-twin/24465>.
- [22] Neznámý. TPV GROUP. *CO JE PDM A PLM | TPV group*. [Online] [Citace: 18. Listopad 2019.] <http://www.tpvgroup.cz/co-je-pdm-a-plm/>.
- [23] Neznámý. PLM Software. *PDM nebo PLM - PLM Software*. [Online] [Citace: 18. Listopad 2019.] <http://www.plmsoftware.cz/pdm-nebo-plm>.
- [24] Stark, John. *Product Lifecycle Management: 21st Century Paradigm for Product Realisation 2nd Edition*. Ženeva : Springer-Verlag London, 2011. ISBN: 978-0-85729-545-3.
- [25] Software, Siemens PLM. *Tecnomatix 12 Plant Simulation 3D*. [Web] 5. Listopad 2014.
- [26] *Simulation of Manufacturing Systems*. LAW, Averill M. a McCOMAS, Michael G. Washington, D.C. : IEEE Computer Society Press, 1997. stránky 86-89. ISBN: 0-7803-5134-7.
- [27] Haltmar, Jaroslav, Maslikiewicz, Ondřej a Froněk, Tomáš. *Vývoj jednoúčelových strojů bez chyb a v krátkém čase*. [Webinář] Praha : Siemens, 28. Únor 2018.
- [28] MCS Software UK. *Simufact - Bending demo for manufacturing simulation*. 23. Červen 2016.
- [29] AXIOM TECH s.r.o. *Process Simulate Human*. [Online] [Citace: 29. Prosinec 2019.] <https://www.axiomtech.cz/25345-process-simulate-human>.
- [30] Advanced Manufacturing, ÅF. *Real Virtual Commissioning by ÅF Part 1*. [Video] 2019.

- [31] Hesselbach, Jürgen a Herrmann, Christoph. *Glocalized Solutions for Sustainability in Manufacturing*. Braunschweig : Springer, 2011. ISBN: 978-3-642-19691-1.
- [32] *SIMIT Simulation Software - SIMIT 9*. Karlsruhe : Siemens AG, 2016.
- [33] SIEMENS. *SIMIT Simulation Platform (V10.1) Operating Manual*. [Dokument] 2019.
- [34] SIEMENS. *SIMIT Simulation V9.0 Getting Started*. [Dokument] 2017.
- [35] *SIMATIC SIMIT (V9.0) - Operating Manual*. Norimberk : Siemens AG, 2016. A5E38238767-AA.
- [36] SIEMENS. *SIMULATION UNIT V9.1 System Manual*. [Dokument] 2019.
- [37] Bakkers, Willy a Siemens Software. *[WHAT'S NEW Simcenter Amesim and Simcenter Sysdm 16]*. [Video] 22. Leden 2018.
- [38] TechSim. TechSim Engineering. *Simcenter Amesim*. [Online] [Citace: 30. Prosinec 2019.] <https://www.techsim.cz/prodej-software-siemens-plm/simcenter-amesim/>.
- [39] SmartMeca. *Simcenter Amesim : modèle de pelle mécanique*. [Video] 23. Únor 2018.
- [40] B+R automatizace, spol. s.r.o. Časopis Automa Využití simulačním experimentů v automatizaci. *Časopis Automa časopis pro automatizační techniku*. [Online] Automa, Květen 2009. [Citace: 20. Listopad 2019.] [http://www.automa.cz/cz/casopis-clanky/vyuziti-simulacnich-experimentu-v-automatizaci-2009\\_05\\_38981\\_6046/](http://www.automa.cz/cz/casopis-clanky/vyuziti-simulacnich-experimentu-v-automatizaci-2009_05_38981_6046/).
- [41] Korendiy, Vitaliy. ResearchGate. *FESTO FluidSim Pneumatics software window*. [Online] Červen 2019. [Citace: 30. Prosinec 2019.] [https://www.researchgate.net/figure/FESTO-FluidSim-Pneumatics-software-window-in-which-the-simulation-of-the-pneumatic\\_fig5\\_336859953](https://www.researchgate.net/figure/FESTO-FluidSim-Pneumatics-software-window-in-which-the-simulation-of-the-pneumatic_fig5_336859953).
- [42] FESTO. *FluidSIM 5 User's Guide*. [Dokument]
- [43] SIEMENS. *Siemens PLM Software: NX*. [Dokument] 2015.
- [44] SIEMENS. Siemens Digital Industries Software. *Siemens Documentation: NX 12.0.2 Help*. [Online] 2018. [Citace: 4. Leden 2020.] [https://docs.plm.automation.siemens.com/tdoc/nx/12.0.2/nx\\_help/#uid:index](https://docs.plm.automation.siemens.com/tdoc/nx/12.0.2/nx_help/#uid:index).
- [45] Lenord, Matthias. *Mechatronics Concept Designer: Quickstart and User Manual*. [Dokument] Los Angeles, Kalifornie, USA : autor neznámý, 12. Květen 2010.
- [46] Katedra priemyselného inžinierstva. *Tecnomatix - Katedra priemyselného inžinierstva*. [Online] [Citace: 5. Leden 2020.] [http://www.priemyselneinzinierstvo.sk/?page\\_id=1546](http://www.priemyselneinzinierstvo.sk/?page_id=1546).
- [47] Neznámý. OnePLM. *Tecnomatix process simulate*. [Online] [Citace: 5. Leden 2020.] <https://oneplm.com/tecnomatix/process-simulate/>.
- [48] Neznámý. Production Modeling Corporation. *Tecnomatix | Process Simulate Robotic*. [Online] [Citace: 5. Leden 2020.]

<https://www.pmc corp.com/Products/SiemensProducts/SiemensTecnomatix/ProcessSimulateRobotics.aspx>.

- [49] SIEMENS. Siemens Digital Industries Software. *Plant Simulation and Throughput Optimization*. [Online] [Citace: 5. Leden 2020.] <https://www.plm.automation.siemens.com/global/en/products/manufacturing-planning/plant-simulation-throughput-optimization.html>.
- [50] Neznámý. Production Modeling Corporation. *Tecnomatix Plant Simulation*. [Online] [Citace: 5. Leden 2020.] <https://www.pmc corp.com/Products/Simulation/TecnomatixPlantSimulation.aspx>.
- [51] PMC Videos. *Plant Simulation With Tecnomatix*. místo neznámé : Springer International Publishing, 2015. ISBN: 978-3-319-19502-5.
- [52] WinMOD. *Virtuelle Inbetriebnahme in der Fertigungstechnik mit WinMOD*. [Video] 2019.
- [53] WinMOD. *Datasheet WinMOD Configuration A780 with PROFIBUS DP and PROFINET-I/O*. [Dokument] Hennigsdorf : autor neznámý, Listopad 2013.
- [54] Visual Components. Visual Components - A world leader in 3D simulation. *Visual Components: 3D manufacturing simulation and ...* [Online] [Citace: 23. Únor 2020.] <https://www.visualcomponents.com/>.
- [55] Visual Components. Visual Components - A world leader in 3D simulation. *Products*. [Online] [Citace: 23. Únor 2020.] <https://www.visualcomponents.com/Products/>.
- [56] Frieß, Michael. ISG: Software solutions and technologies for automation and ... *Effizienter Projektverlauf mit Robotersimulation und Virtueller Inbetriebnahme*. [Online] 1. Duben 2014. [Citace: 9. Březen 2020.] [https://www.isg-stuttgart.de/fileadmin/user\\_upload/virtuos/Praesentation\\_Erhardt\\_Abt.pdf](https://www.isg-stuttgart.de/fileadmin/user_upload/virtuos/Praesentation_Erhardt_Abt.pdf).
- [57] ISG Industrielle Steuerungstechnik. ISG: Software solutions and technologies for automation and ... *Simulation / Digitale Zwillinge Reale Inbetriebnahme an virtuellen Modellen*. [Online] 20. Listopad 2019. [Citace: 9. Březen 2020.] [https://www.isg-stuttgart.de/fileadmin/user\\_upload/virtuos/ISG-virtuos-Simulationsplattform-2019\\_engl.pdf](https://www.isg-stuttgart.de/fileadmin/user_upload/virtuos/ISG-virtuos-Simulationsplattform-2019_engl.pdf).
- [58] ISG Industrielle Steuerungstechnik. ISG: Software solutions and technologies for automation and ... *ISG-virtuos: Info*. [Online] [Citace: 9. Březen 2020.] [https://www.isg-stuttgart.de/fileadmin/user\\_upload/virtuos/2019\\_11\\_14\\_Broschuere\\_virtuos\\_2019\\_EN\\_ANSIC HT.pdf](https://www.isg-stuttgart.de/fileadmin/user_upload/virtuos/2019_11_14_Broschuere_virtuos_2019_EN_ANSIC HT.pdf).
- [59] Böhler, Tino. Produktion Online. *Einfache Konfiguration komplexer Anlagen*. [Online] 17. Červen 2016. [Citace: 9. Březen 2020.] <https://www.produktion.de/technik/einfache-konfiguration-komplexer-anlagen-126.html>.
- [60] CAD/CAM GROUP. CAD/CAM GROUP. *DELMIA V5*. [Online] [Citace: 9. Březen 2020.] <https://www.cadcam-group.eu/products/delmia-v5>.

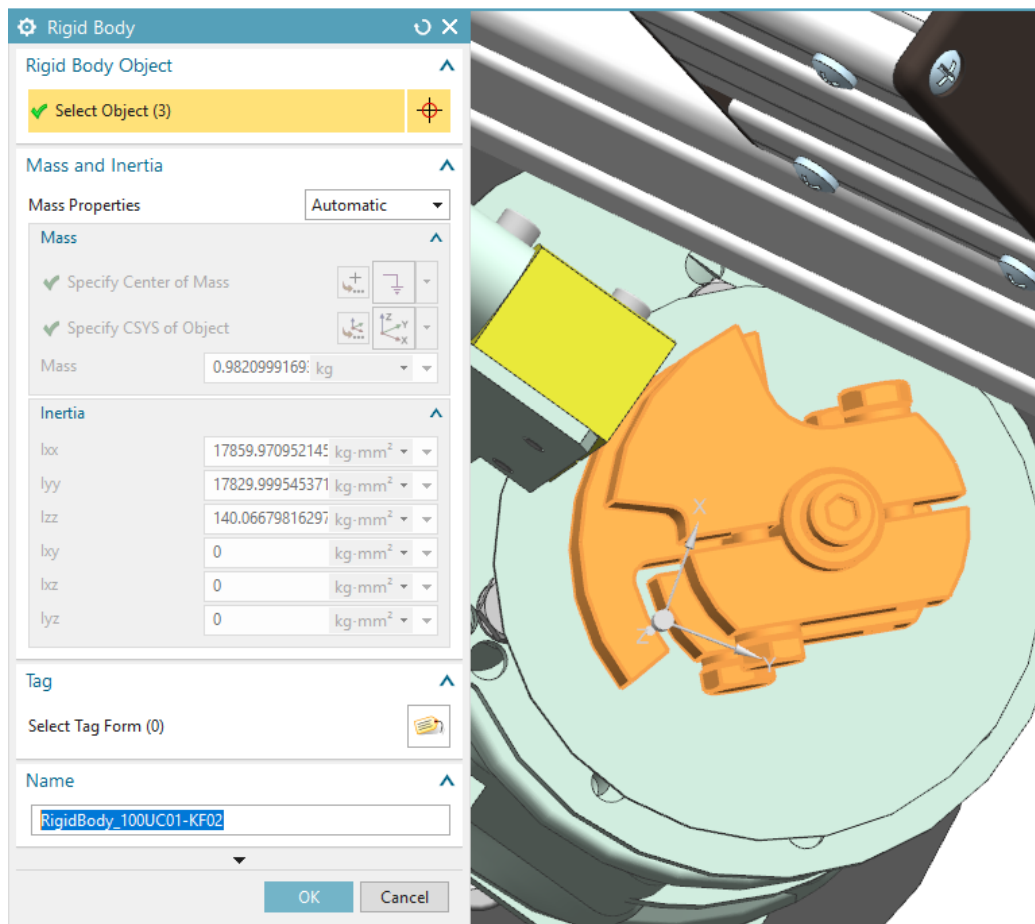


- [61] InFlow Technology. InFlow Technology - PLM / PDM Services. *DELMIA Virtual Commissioning Expert (RCV) Role - InFlow ...* [Online] [Citace: 9. Březen 2020.] <https://www.inflow-tech.com/solutions/delmia/roles/virtual-commissioning-expert-rcv/>.
- [62] Karras, U. *FESTO, CIROS Studio 1.0: User's Guide*. [Dokument] Srpen 2008.
- [63] Festo Japan. *CIROS Robot Simulation*. [Video] 11. Srpen 2017.
- [64] Factory I/O. Next-Gen PLC Training - Factory I/O. *Factory I/O - Documentation*. [Online] [Citace: 10. Březen 2020.] <https://docs.factoryio.com/>.
- [65] Robert, L. *Factory I/O Assembly Process*. 16. Červenec 2016. Sv. Video.
- [66] SEW Eurodrive. *Variable Speed Gearmotors Catalog*. [Dokument] Srpen 2004. 11230126 / EN.
- [67] Tramec. *WORM GEARBOXES SERIES SFK - SRK - BFK - BRK*. [Dokument] Únor 2020.
- [68] SEW. *Project Planning of Drives*. [Dokument] Říjen 2001.
- [69] Control Tutorials for MATLAB and Simulink. *Control Tutorials for MATLAB and Simulink - Motor speed: Simulink Modeling*. [Online] [Citace: 11. Duben 2020.] <http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SimulinkModeling>.
- [70] FESTO. *Toothed belt axes ELGR*. [Dokument] Srpen 2019.
- [71] FESTO. *Krokové motory EMMS-ST*. Listopad 2016.
- [72] FESTO. *Description FHPP*. [Dokument] 2009.
- [73] FESTO. *CMMO-ST-C5-1-LKP*. [Dokument]
- [74] SIEMENS. *SIMATIC S7-PLCSIM Advanced: Co-Simulation via API*. [Dokument] Prosinec 2016.

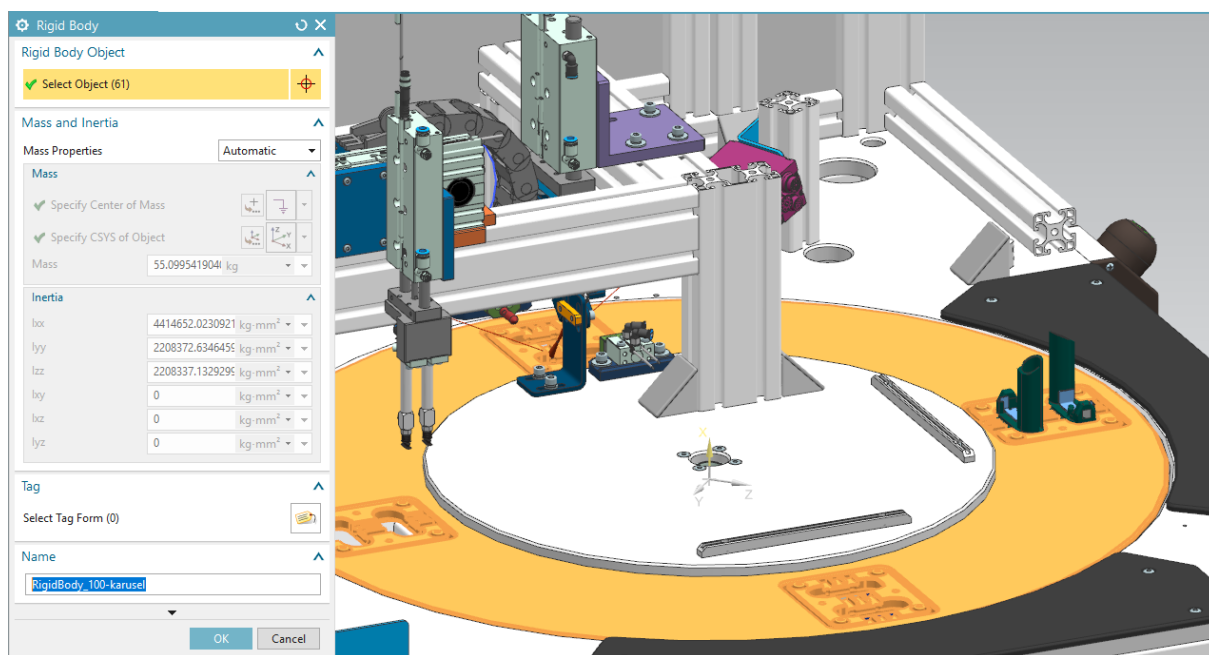
## Seznam příloh

Příloha A:	<i>Fyzikálně-kinematický model v NX MCD</i> .....	I
Příloha B:	<i>Elektrický a behaviorální model v SIMIT</i> .....	XIII
Příloha C:	<i>Návrh řídicí aplikace v TIA Portal</i> .....	XIX
Příloha D:	<i>Komunikační aplikace</i> .....	XXXV
Příloha E:	<i>Virtuální zprovoznění stroje</i> .....	XLII
Příloha F:	<i>CD-ROM</i> .....	XLVIII

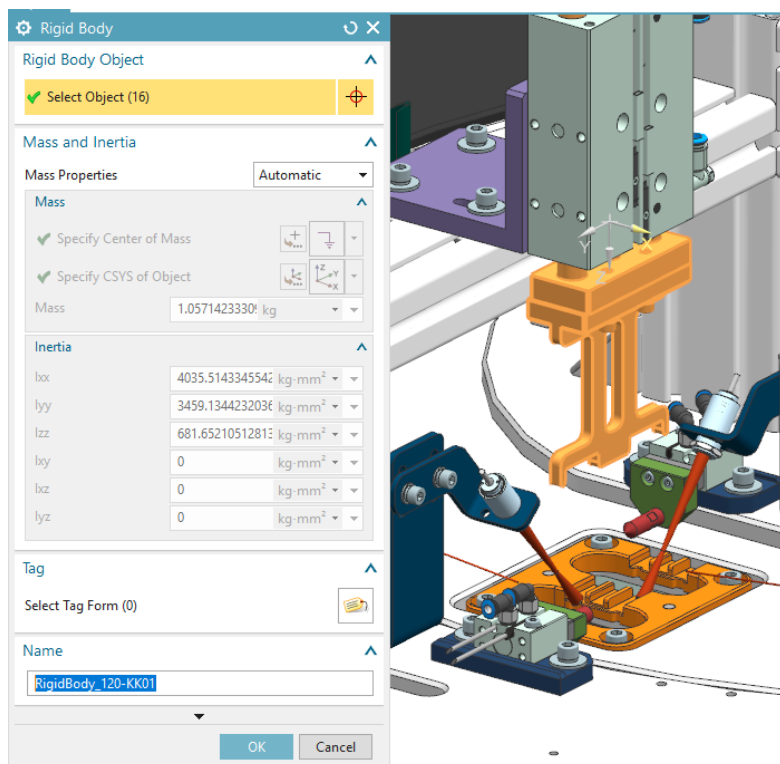
Příloha A: *Fyzikálně-kinematický model v NX MCD*



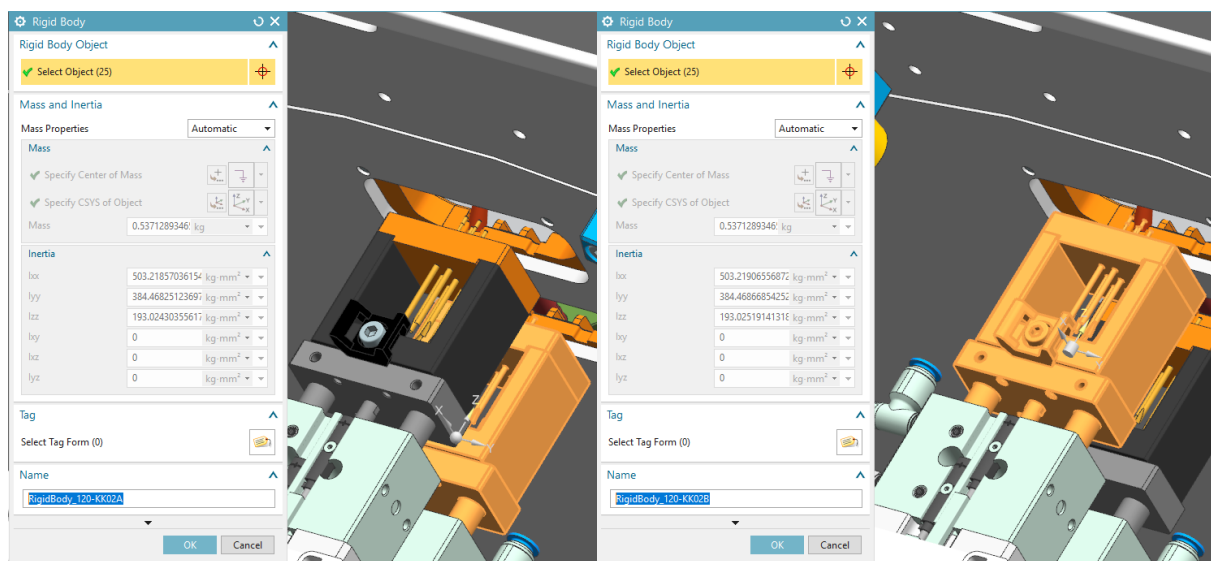
Obrázek A.1: *Nedeformovatelné těleso hřídele*



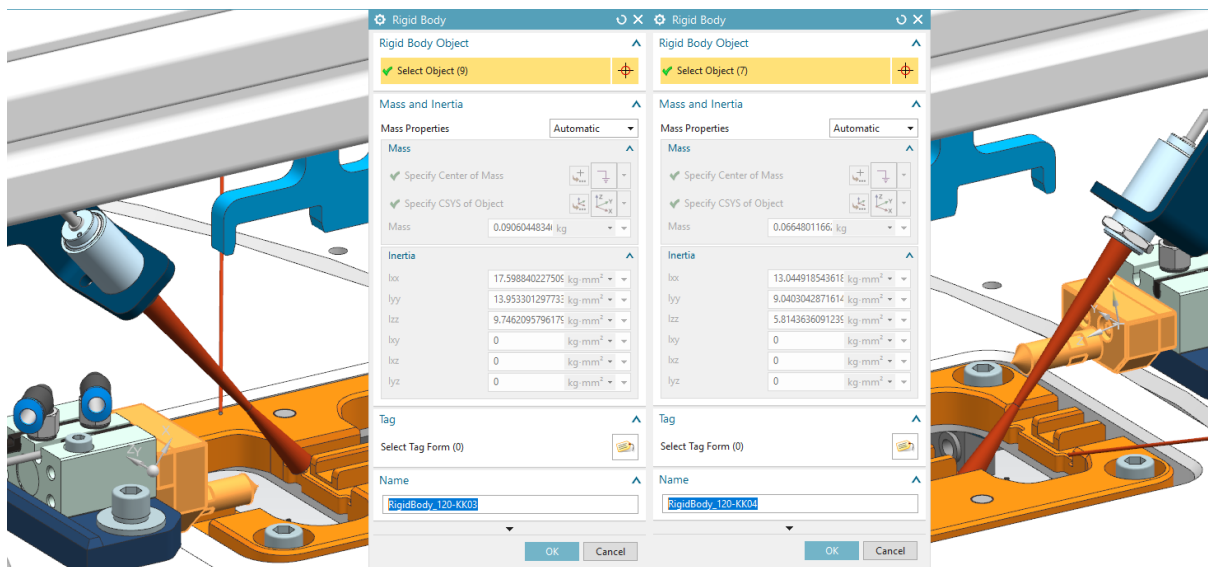
Obrázek A.2: *Nedeformovatelné těleso karuselu*



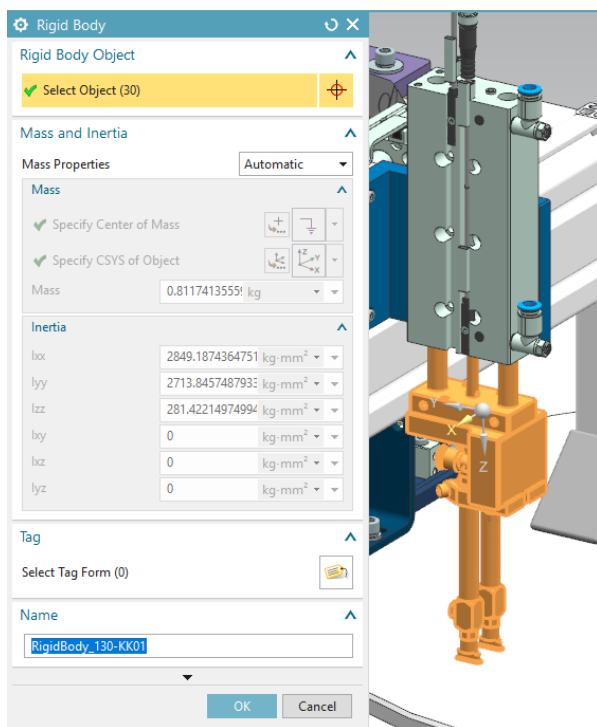
Obrázek A.3: Nedeformovatelné těleso přidržovacího válce



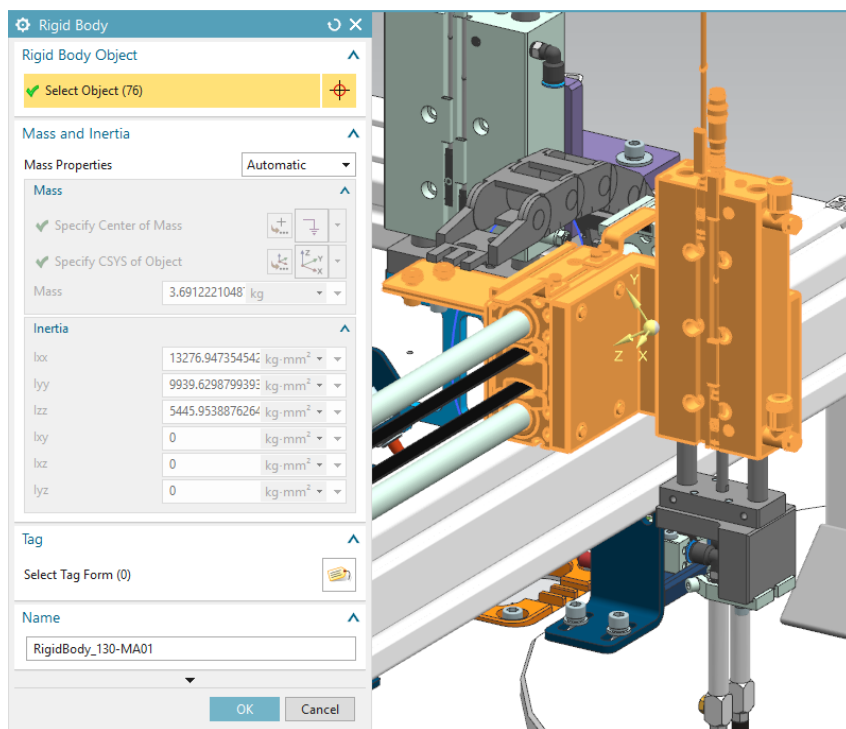
Obrázek A.4: Nedeformovatelná tělesa válců kontaktních jednotek



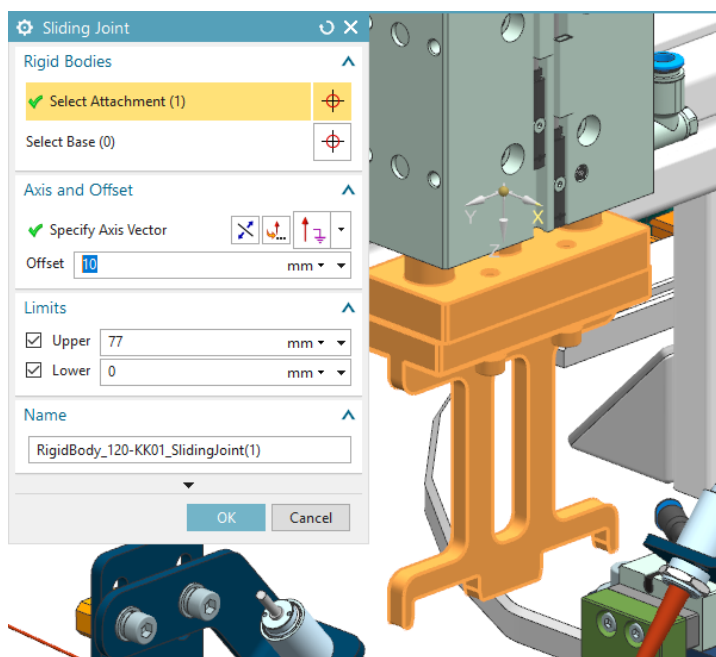
Obrázek A.5: *Nedeformovatelná tělesa značících válců*



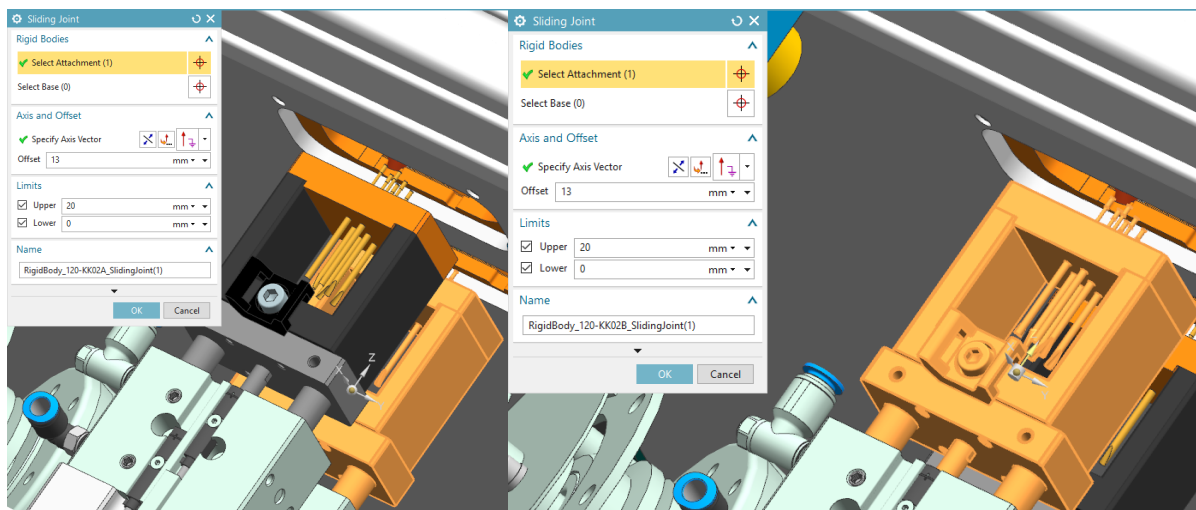
Obrázek A.6: *Nedeformovatelné těleso válce manipulátoru v ose Z*



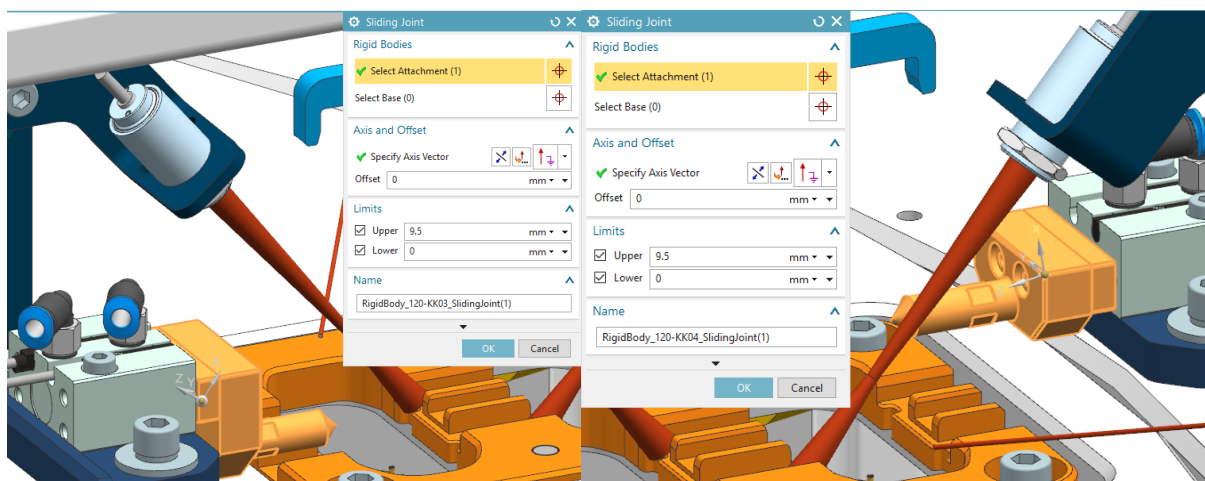
Obrázek A.7: *Nedeformovatelné těleso manipulátoru v ose X*



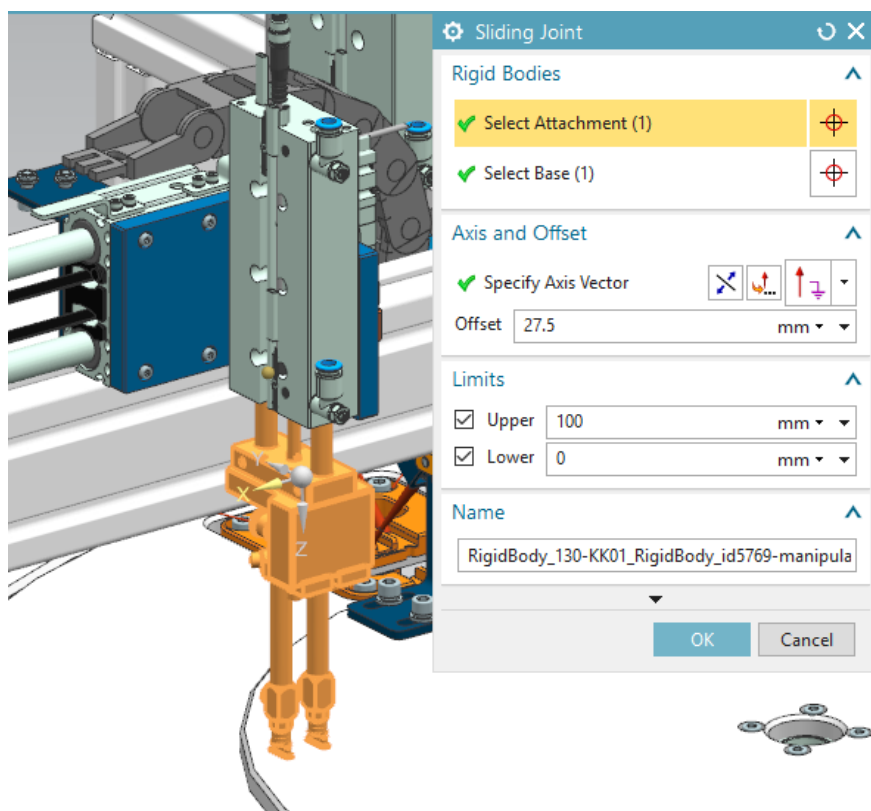
Obrázek A.8: *Posuvný spoj přidržovacího válce*



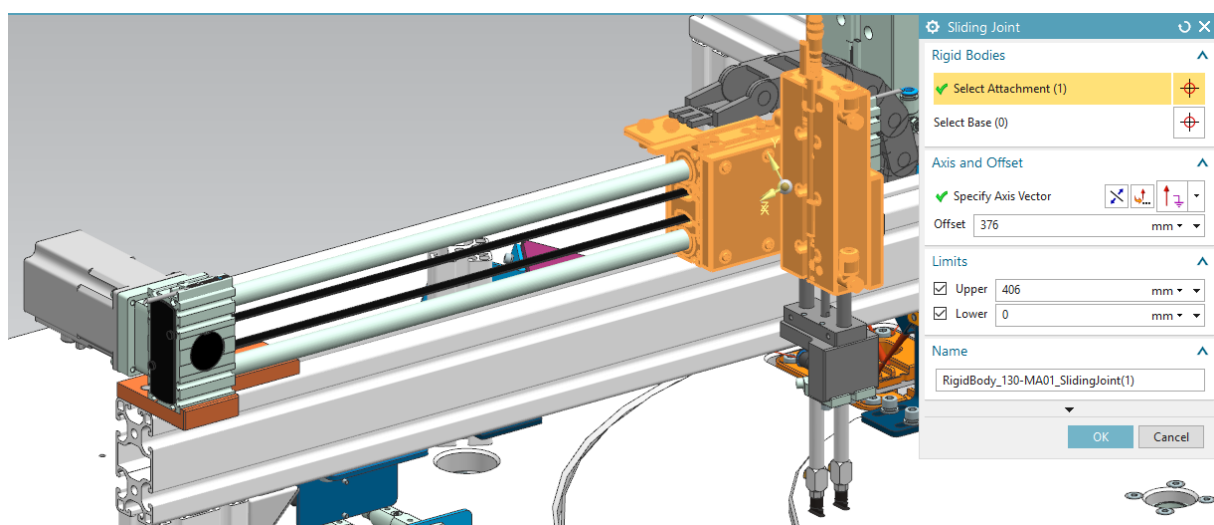
Obrázek A.9: Posuvné spoje válců kontaktních jednotek



Obrázek A.10: Posuvné spoje značících válců

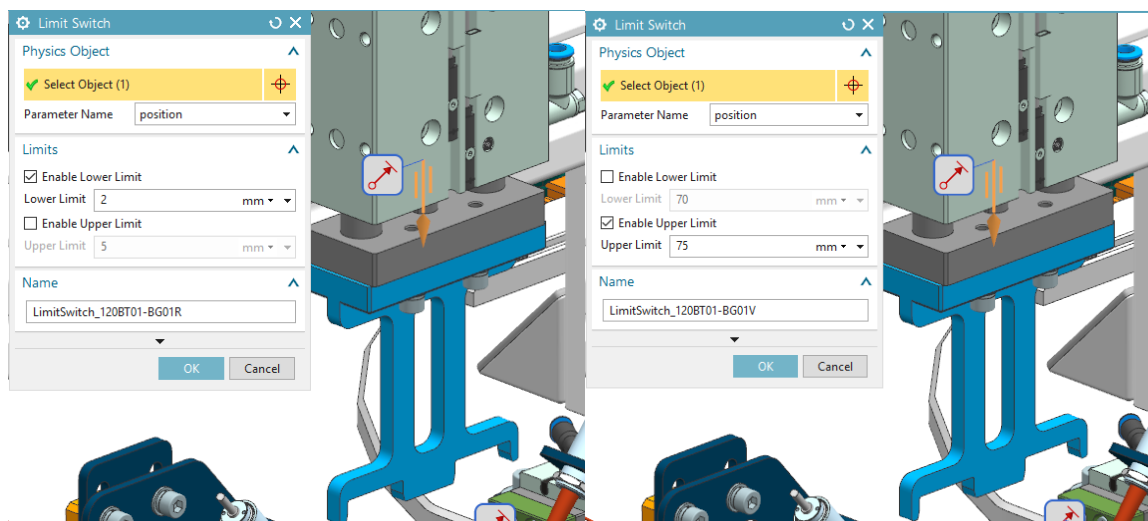


Obrázek A.11: Posuvný spoj válce manipulátoru v ose Z

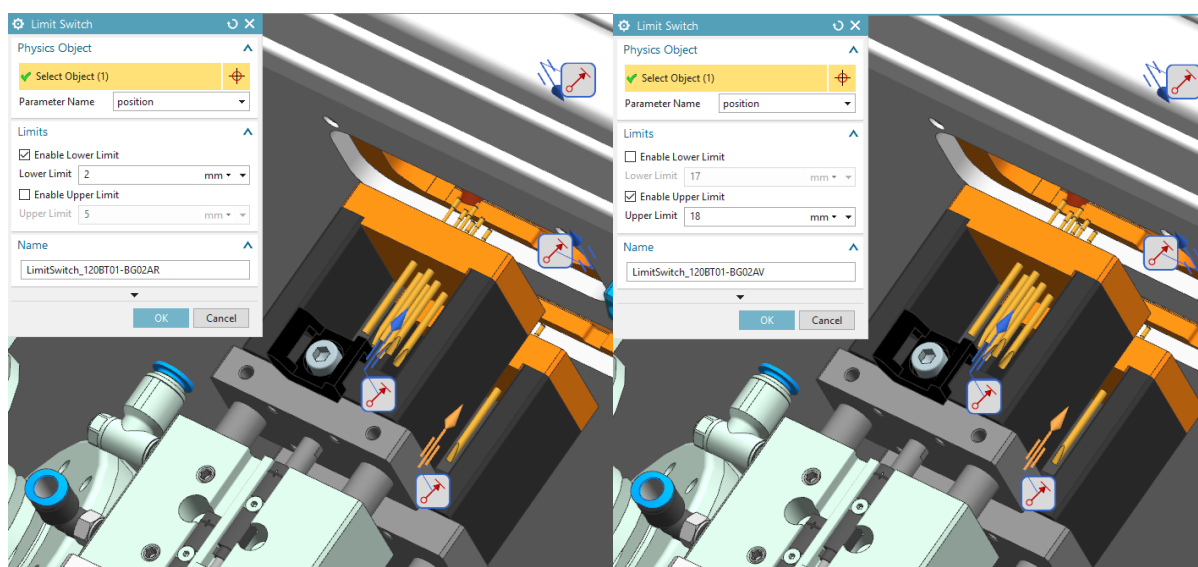


Obrázek A.12: Posuvný spoj manipulátoru v ose X

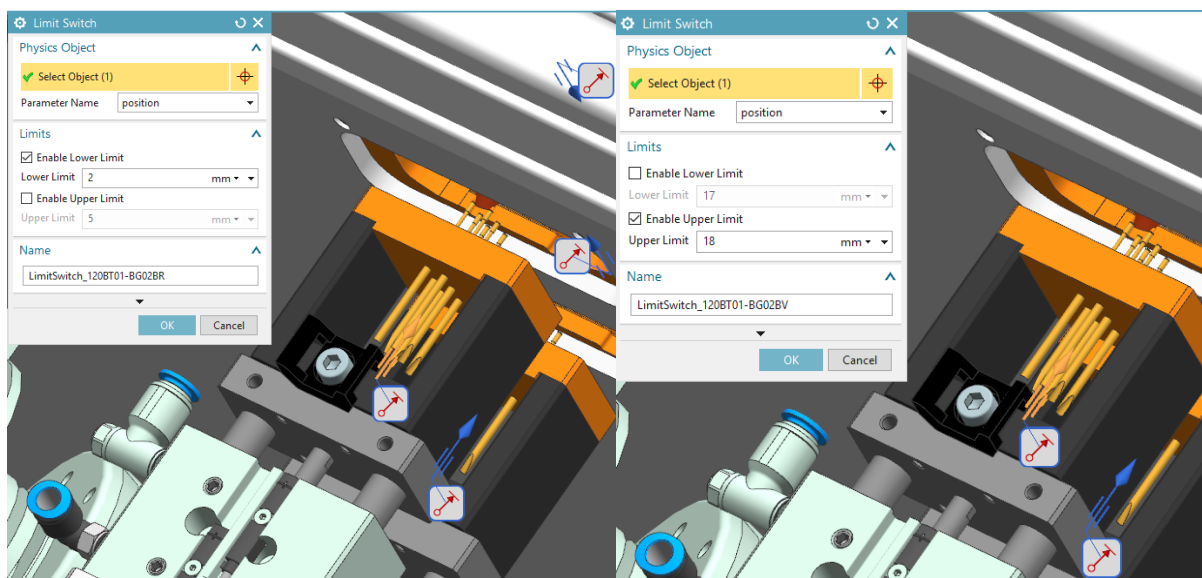




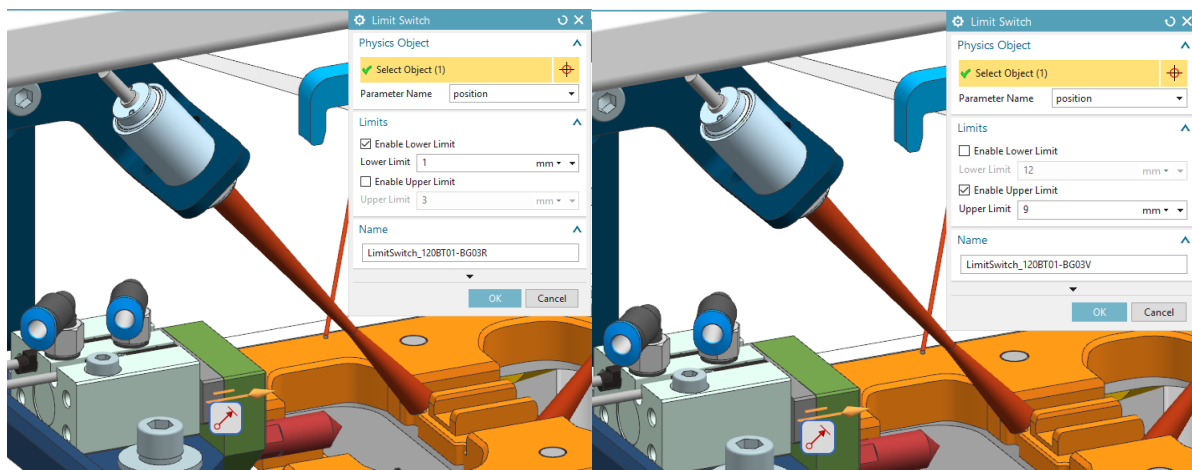
Obrázek A.13: *Koncové spínače přidržovacího válce*



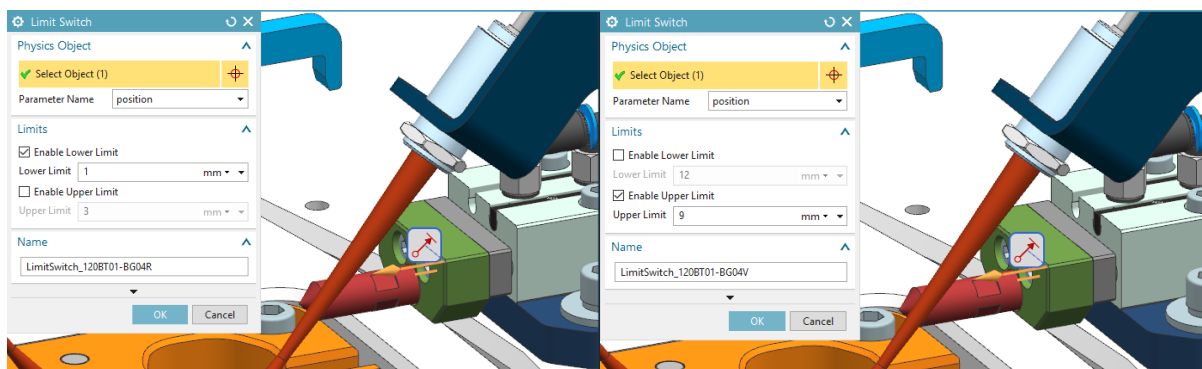
Obrázek A.14: *Koncové spínače válce kontaktních jednotek A*



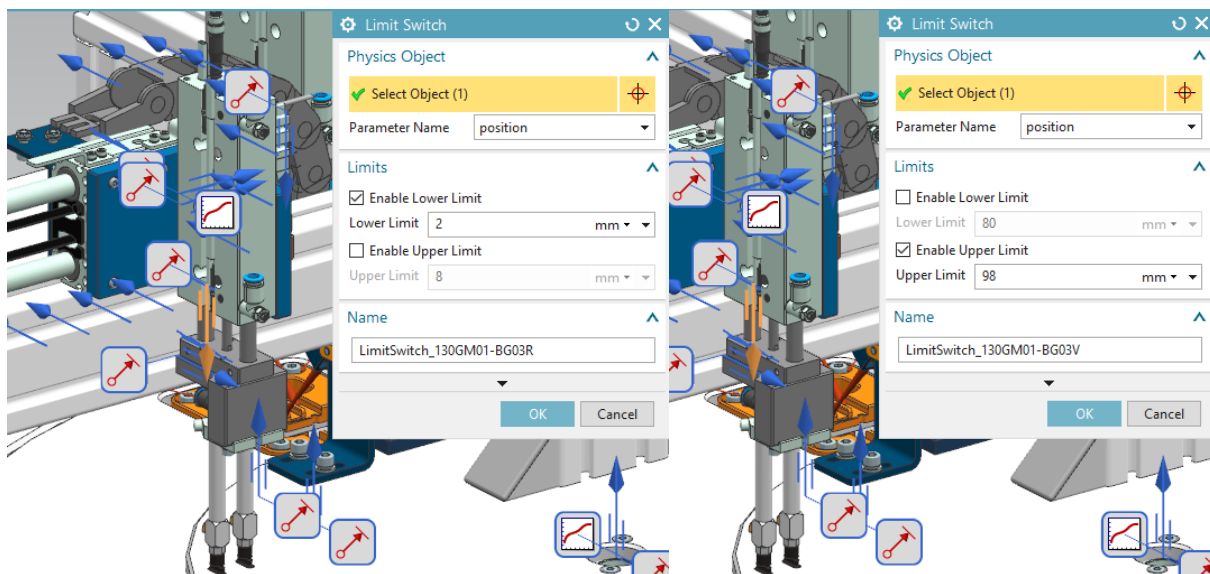
Obrázek A.15: *Koncové spínače válce kontaktních jednotek B*



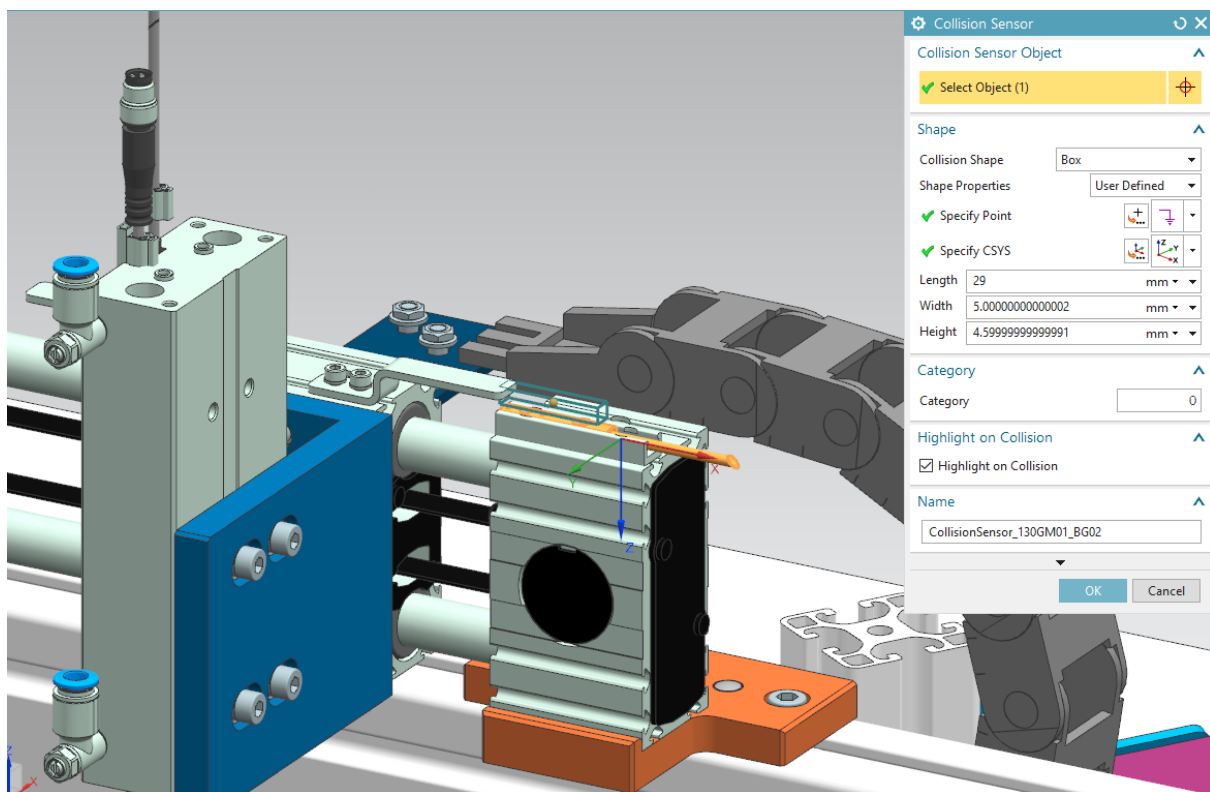
Obrázek A.16: *Koncové spínače značícího válce A*



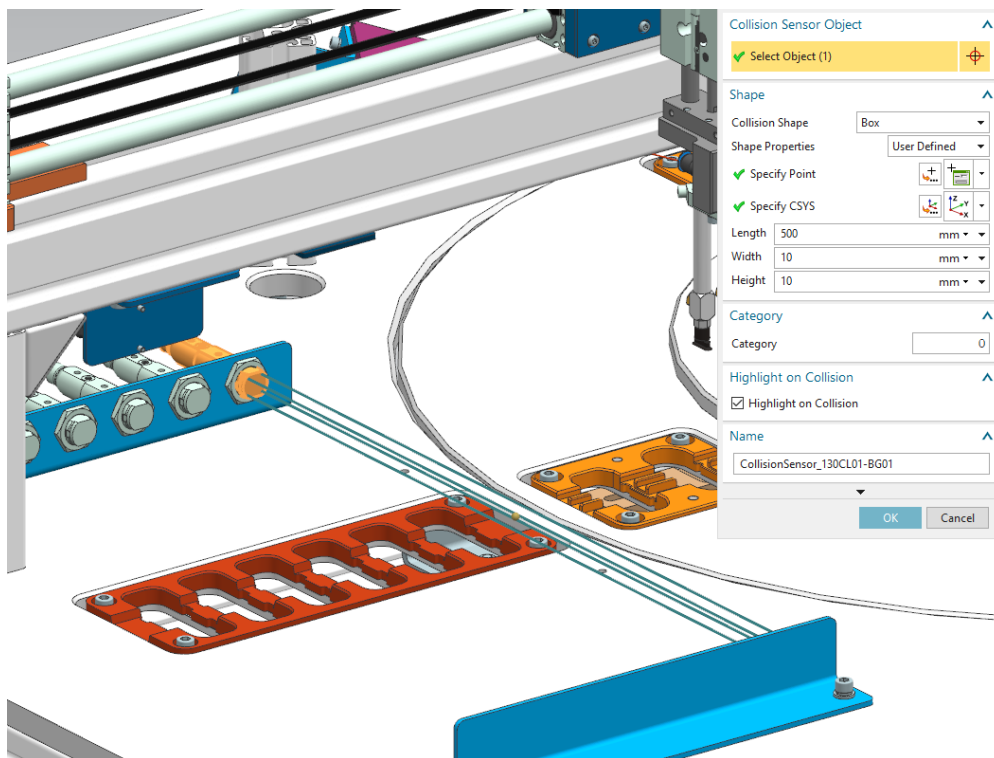
Obrázek A.17: *Koncové spínače značícího válce B*



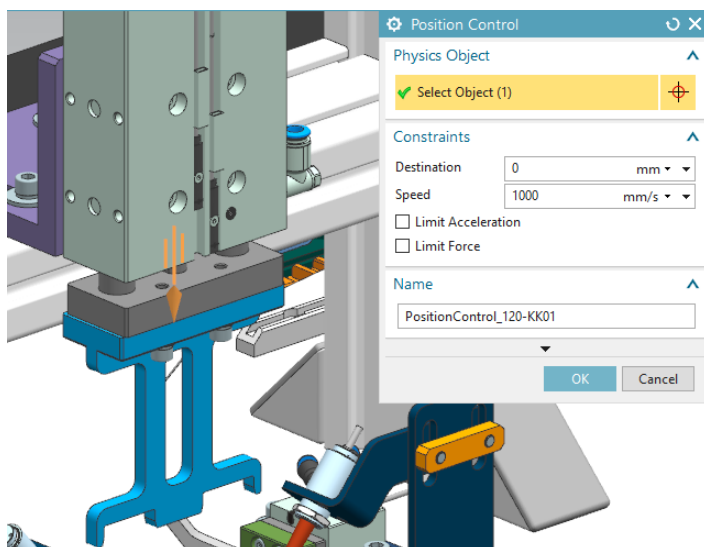
Obrázek A.18: *Koncové spínače válce manipulátoru v ose Z*



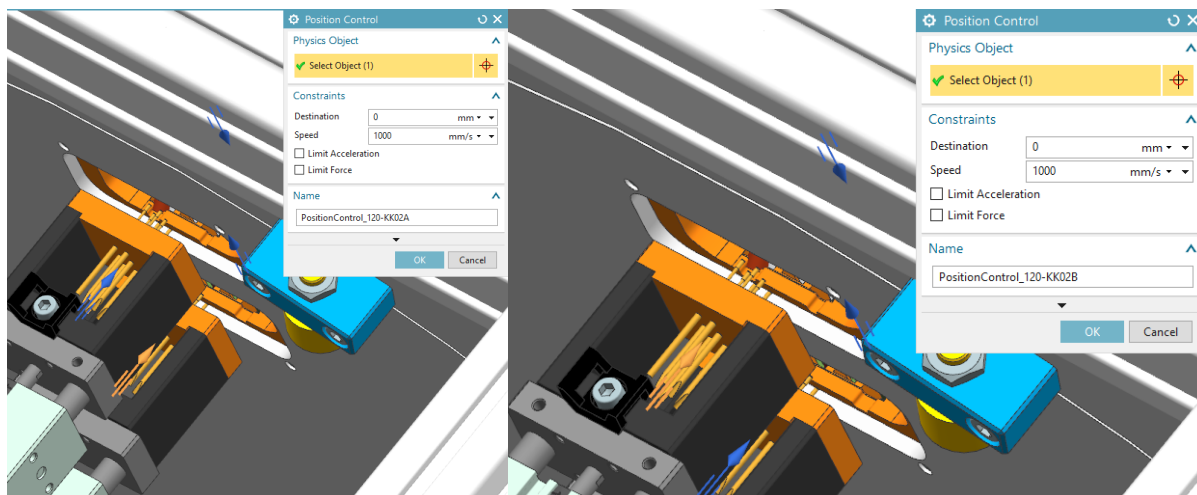
Obrázek A.19: *Kolizní senzor koncové polohy manipulátoru v ose X*



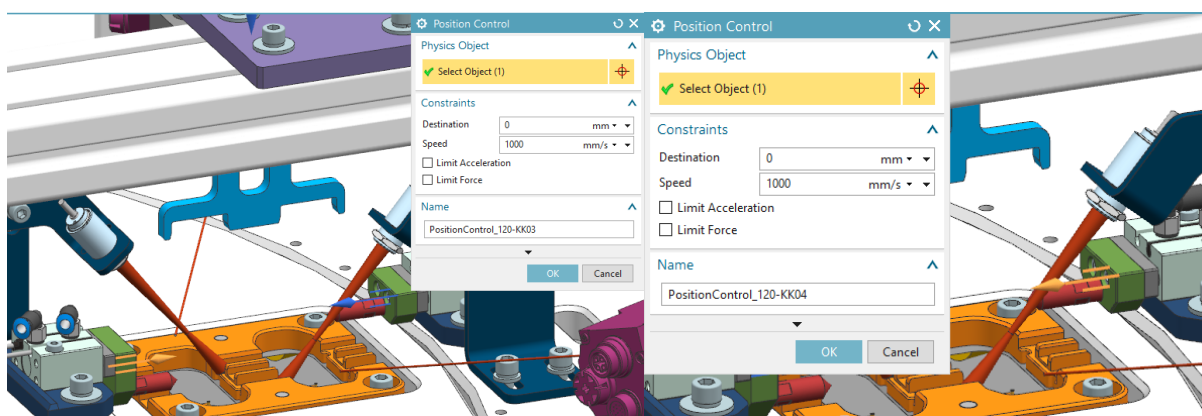
Obrázek A.20: Kolizní senzor NOK lůžka



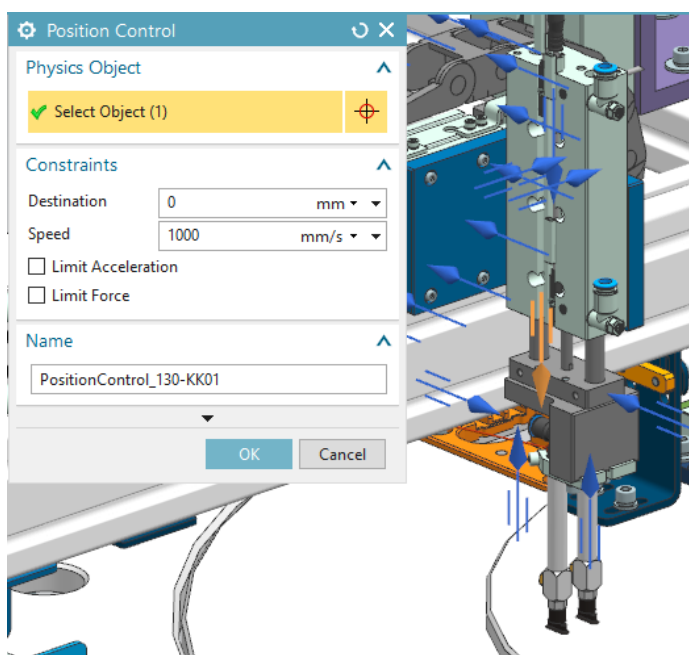
Obrázek A.21: Řízení polohy přidržovacího válce



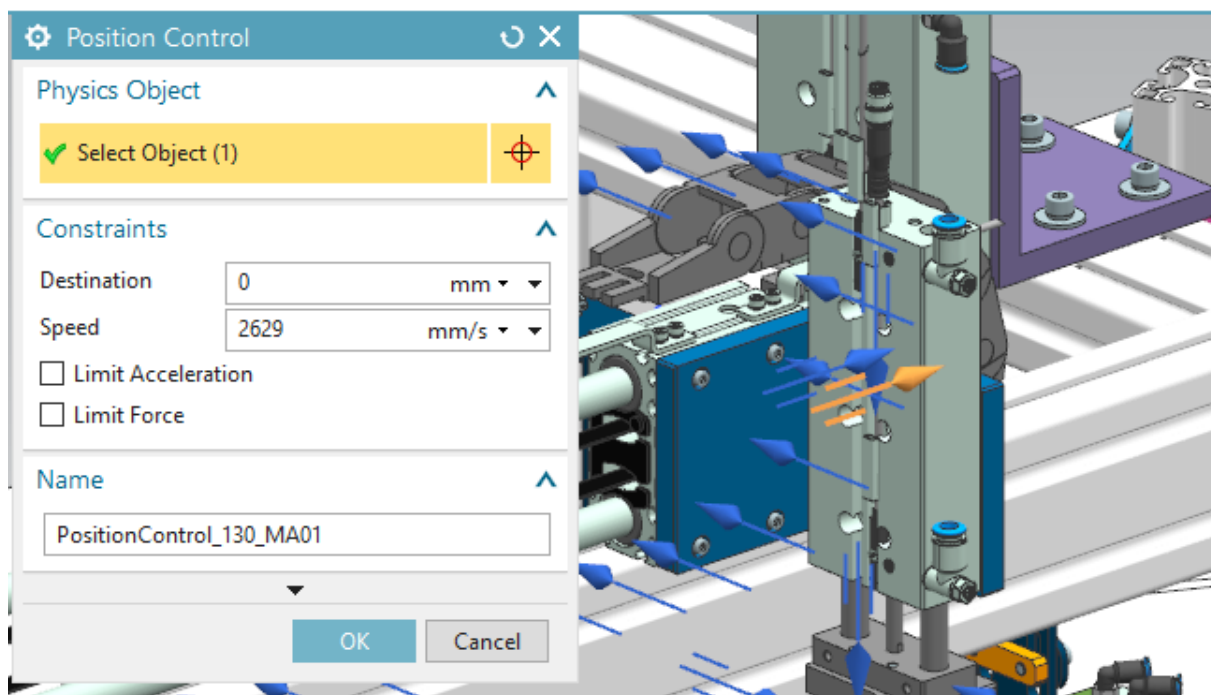
Obrázek A.22: Řízení polohy válců kontaktních jednotek



Obrázek A.23: Řízení polohy značících válců



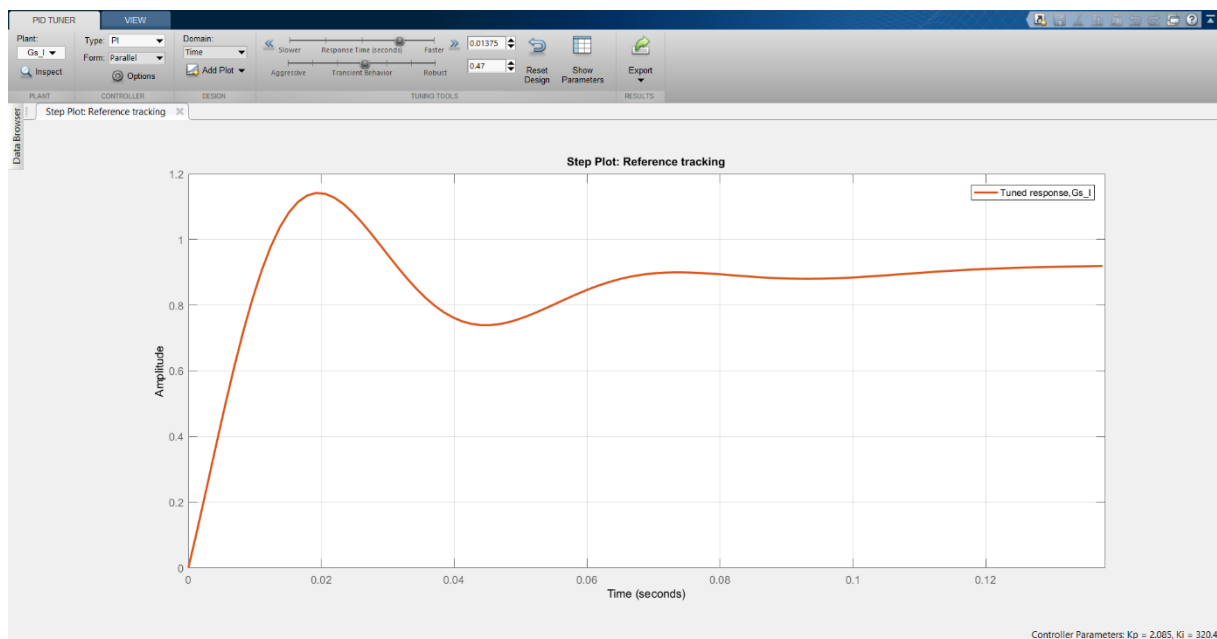
Obrázek A.24: Řízení polohy válce manipulátoru v ose Z



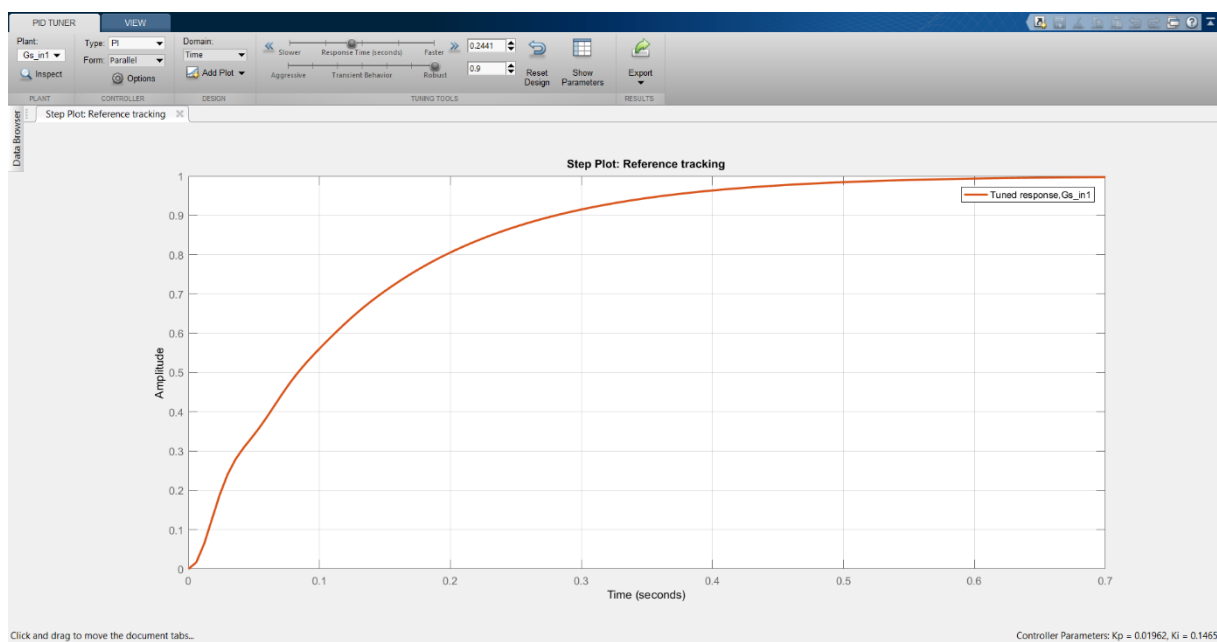
Obrázek A.25: Řízení polohy manipulátoru v ose  $X$



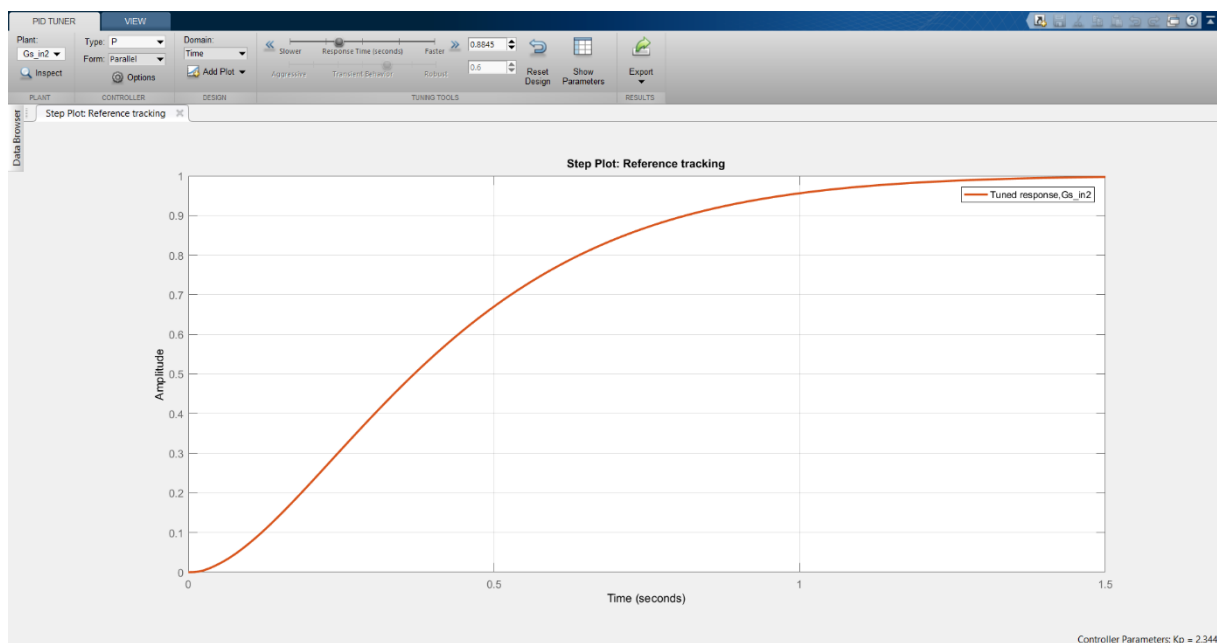
Příloha B: *Elektrický a behaviorální model v SIMIT*



Obrázek B.1: *Nastavení PI regulátoru proudu v PID Tool pro CMMO jednotku*

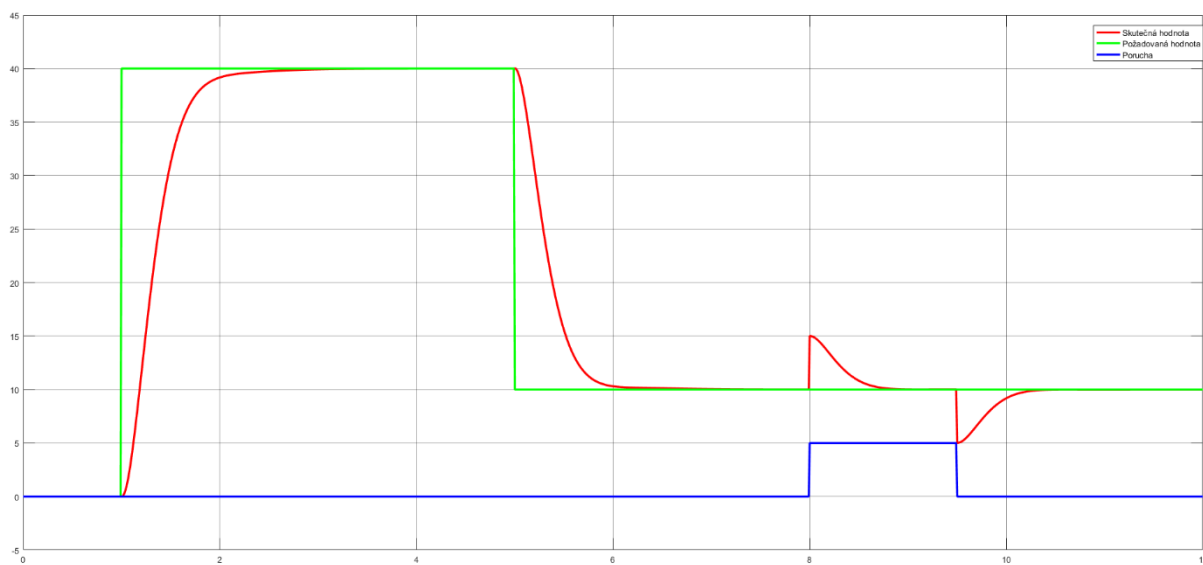


Obrázek B.2: *Nastavení PI regulátoru rychlosti v PID Tool pro CMMO jednotku*



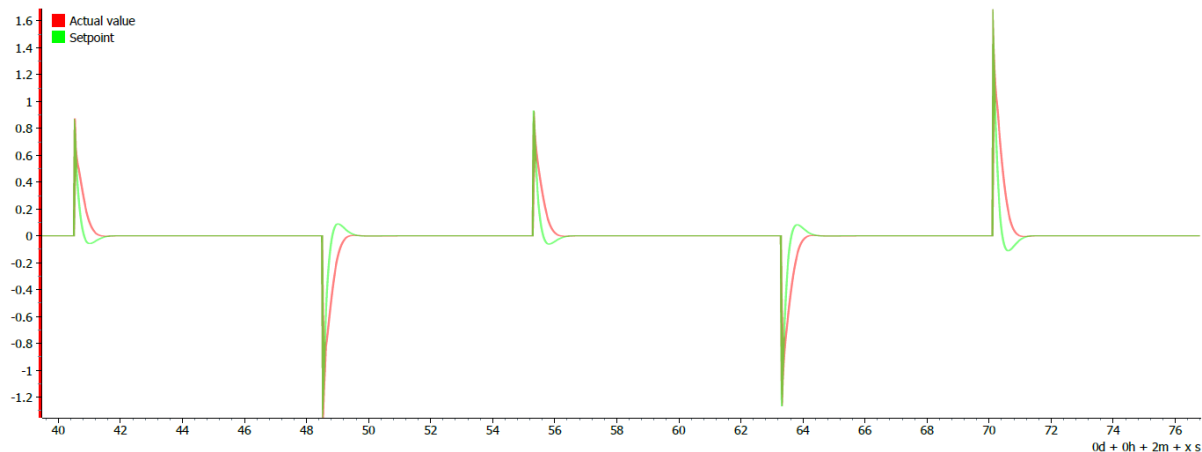
Obrázek B.3: *Nastavení P regulátoru polohy v PID Tool pro CMMO jednotku*

Obrázek B.4: *Model DC motoru v prostředí MATLAB Simulink*

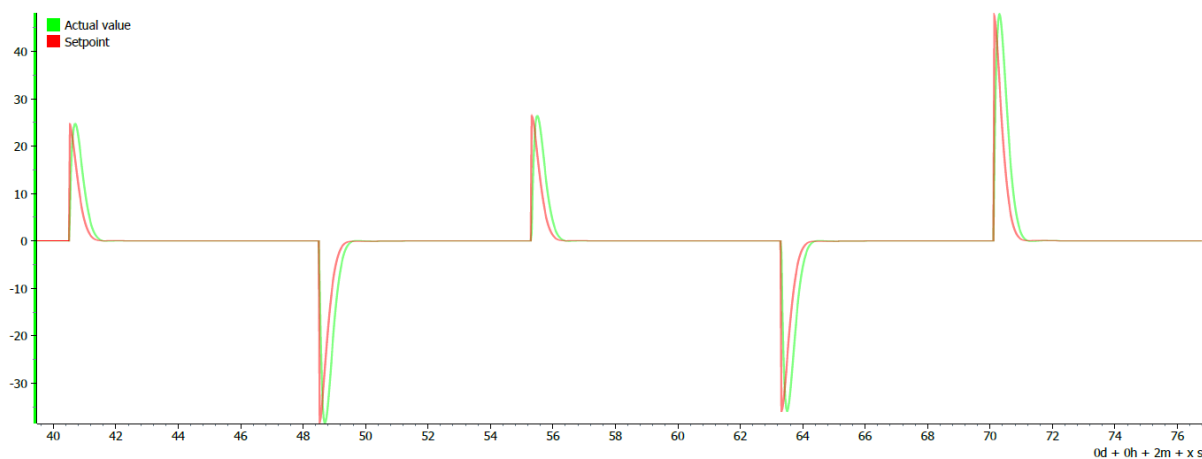


Obrázek B.5: *Regulace polohy v prostředí MATLAB Simulink*

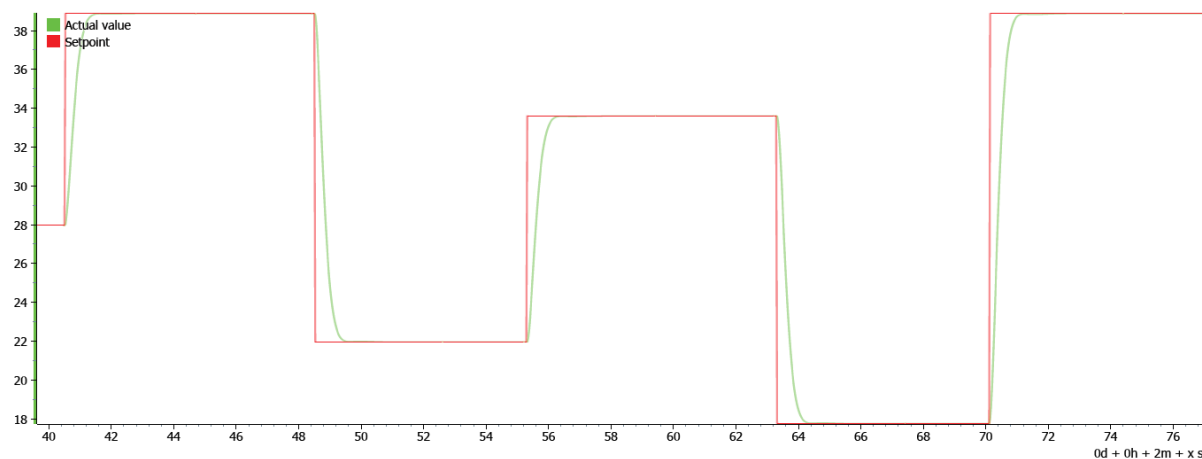




Obrázek B.6: *Regulace proudu tekoucího do DC motoru v SIMIT*



Obrázek B.7: *Regulace rychlosti DC motoru v SIMIT*



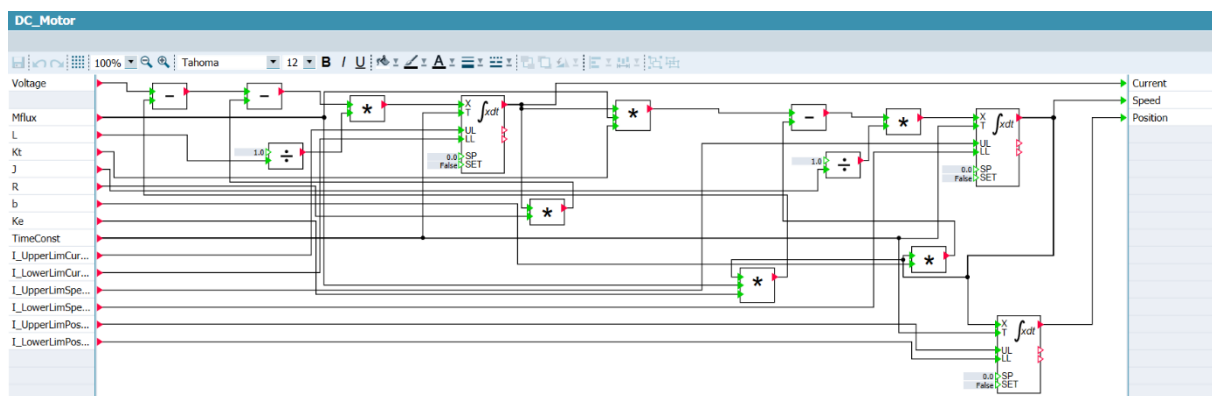
Obrázek B.8: *Regulace polohy DC motoru v SIMIT*

```

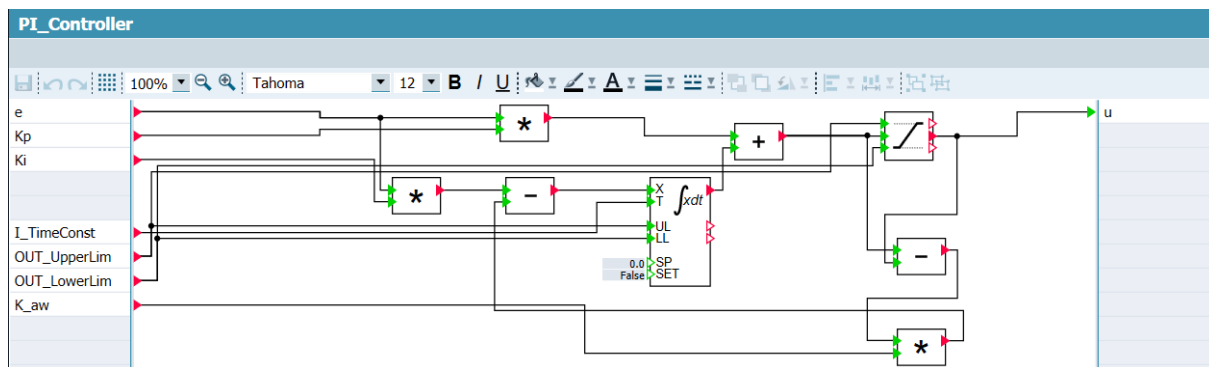
1 - s=tf('s')
2 - %Parametry DC motoru
3 - L=5*0.5*10^-3 %50*
4 - Kt=0.16
5 - Ke=0.16663
6 - J=300.1935*10^-6
7 - R=0.15
8 - b=2.85*10^-3
9 - flux=1;
10
11 %P?enos soustavy pro proud
12 Gs_I=1/((L*s+R)+(Ke*Kt*flux*flux)/(J*s+b))
13 %P?enos soustavy pro rychlost
14 Gs_om=(Kt*flux)/((L*s+R)*(J*s+b)+(Ke*Kt*flux*flux))
15 %P?enos soustavy pro pozici
16 Gs_th=(Kt*flux)/(s*((L*s+R)*(J*s+b)+(Ke*Kt*flux*flux)))
17
18 %Návrh regulátoru proudu pro soustavu Gs_I
19 pidtool
20 %P?echodová char soustavy pro proud
21 figure
22 step(Gs_I)
23
24 %Regulátor proudu
25 Kp_I=0.004702862274463
26 Ki_I=33.571906641818860
27 %Kp_I=C1.Kp
28 %Ki_I=C1.Ki
29 Gr_I=Kp_I+Ki_I/s
30
31 %Návrh regulátoru rychlosti pro p?enos Gs_in1
32 Gs_in1=(Gr_I*Gs_om)/(1+(Gr_I*Gs_I))
33 pidtool
34 %P?echodová char soustavy pro vnit?ní p?enos soustavy Gs_in1
35 figure
36 step(Gs_in1)
37
38 %Regulátor rychlosti
39 Kp_om=0.064624578075298
40 Ki_om=0.193869679947667
41 %Kp_om=C2.Kp
42 %Ki_om=C2.Ki
43 Gr_om=Kp_om+Ki_om/s
44
45 %Návrh regulátoru polohy na soustavu Gs_in2
46 Gs_in2=(Gr_om*((Gr_I*Gs_th)/(1+(Gr_I*Gs_I)))/(1+(Gr_om*Gs_in1))
47 pidtool
48 %P?echodová char soustavy pro vnit?ní p?enos soustavy Gs_in2
49 figure
50 step(Gs_in2)
51
52 %Regulátor pozice
53 Kp_th=3.172033322605294
54 Ki_th=0
55 %Kp_th=C3.Kp
56 %Ki_th=C3.Ki
57 Gr_th=Kp_th+Ki_th/s

```

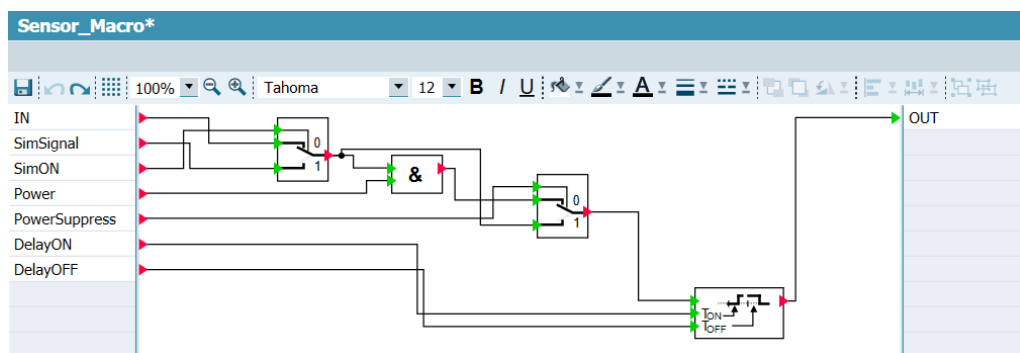
Obrázek B.9: Kód pro návrh regulátorů v MATLABU pomocí nástroje PID Tool



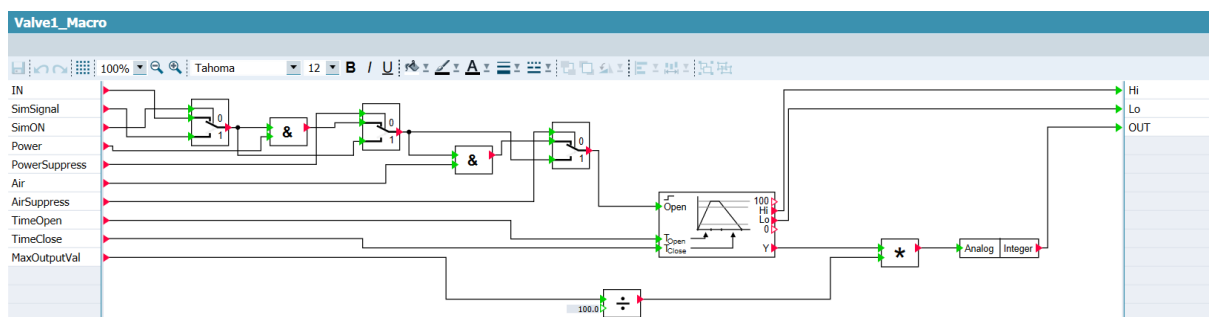
Obrázek B.10: Makro modelu DC motoru



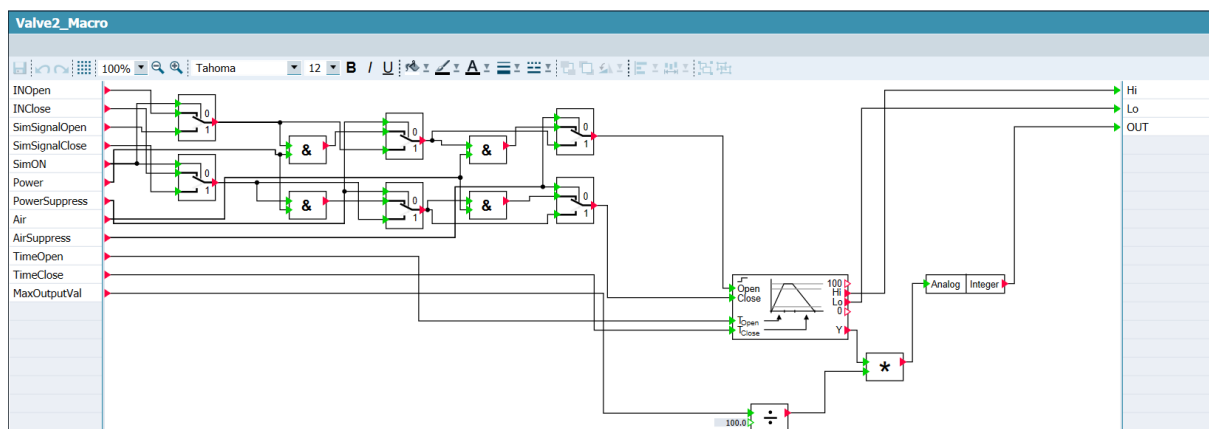
Obrázek B.11: Makro modelu PI regulátoru



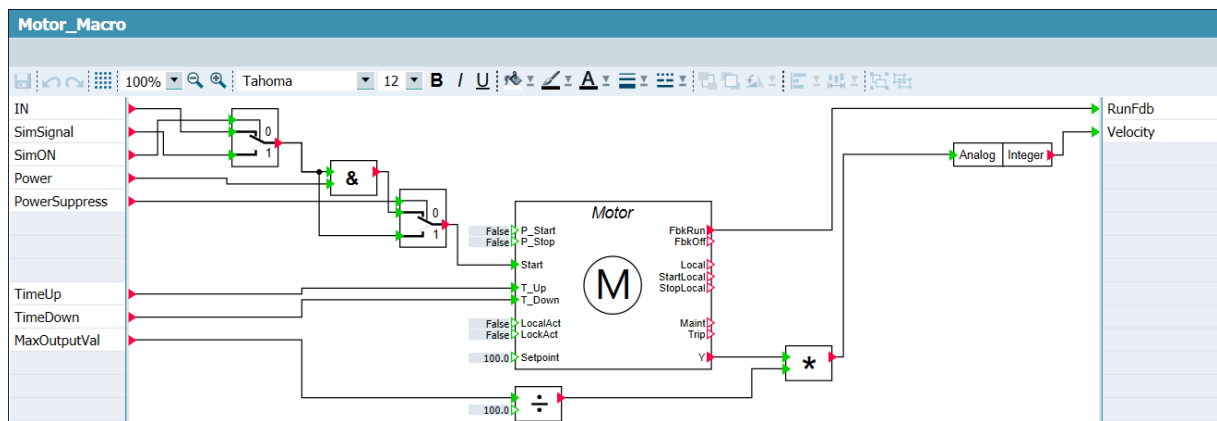
Obrázek B.12: Makro senzoru



Obrázek B.13: Makro dvoupolohového monostabilního ventilu



Obrázek B.14: Makro třípolohového bistabilního ventilu



Obrázek B.15: Makro motoru karuselu

Příloha C: *Návrh řídicí aplikace v TIA Portal*

STATUS_100										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static									
2	xxxxx_BinarySensors...	Byte	16#0							
3	100GL01-BG01	Bool	false							Sensor turn table stop
4	100GL01-BG02	Bool	false							Sensor turn table in position
5	100UR01-BP01_1	Bool	false							Sensor work pressure out 1
6	100UR01-BP01_2	Bool	false							Sensor work pressure out 2
7	xxxxx_BinaryOthers...	Byte	16#0							
8	UC01-QA01	Bool	false							Motor protect for turn table OK
9	UC01-FC03	Bool	false							Power supply 5V OK
10	UC01-FB21	Bool	false							Light and plugs OK
11	UC01-FB22	Bool	false							SSI plugs OK
12	UH01-FC30	Bool	false							Protect 24VDC OK
13	UR01-SF03	Bool	false							Button Start
14	UR01-SF04	Bool	false							Button reset
15	UF01-SF01	Bool	false							Button steps
16	UF01-SF01_13	Bool	false							Button steps is conect
17	UH01-TF01	Bool	false							DUT1 test OK
18	UH01-TF02	Bool	false							DUT2 test OK
19	UH01-TF03	Bool	false							SSI test done
20	UH01-TF04	Bool	false							DUT1 write result OK
21	UH01-TF05	Bool	false							DUT2 write result OK
22	UH01-TF06	Bool	false							Write test done
23	UH01-TF07	Bool	false							Watch dog
24	xxxxx_Drives_xxxxx	Byte	16#0							
25	MA01	Struct								Motor turn table
26	GL01-MA01Drive1	Struct								Drive of motor turn table
27	xxxxx_Others_xxxxx	Byte	16#0							
28	PosTurnTable	Bool	false							Turn table is in position
29	TurnTableRun	Bool	false							Turn table run
30	xxxxx_MatCard_xox...	Byte	16#0							

Obrázek C.1: *Statusový DB stanice 100*

HMI_PARAM_100										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static									
2	xxxxx_BinarySensors...	Byte	16#0							
3	GL01-BG01	*UDT_CH_DI_Param...								Turn table stop
4	GL01-BG02	*UDT_CH_DI_Param...								Turn table position
5	UR01-BP01_1	*UDT_CH_DI_Param...								Pressure sensor out 1
6	UR01-BP01_2	*UDT_CH_DI_Param...								Pressure sensor out 2
7	xxxxx_BinaryOthers...	Byte	16#0							
8	UC01-QA01	*UDT_CH_DI_Param...								Motor protect for turn table OK
9	UC01-FC03	*UDT_CH_DI_Param...								Power supply 5V OK
10	UC01-FB21	*UDT_CH_DI_Param...								Light and plugs OK
11	UC01-FB22	*UDT_CH_DI_Param...								SSI plugs OK
12	UH01-FC30	*UDT_CH_DI_Param...								Protect 24VDC OK
13	UR01-SF03	*UDT_CH_DI_Param...								Button Start
14	UR01-SF04	*UDT_CH_DI_Param...								Button reset
15	UF01-SF01	*UDT_CH_DI_Param...								Button steps
16	UF01-SF01_13	*UDT_CH_DI_Param...								Button steps is conect
17	UH01-TF01	*UDT_CH_DI_Param...								DUT1 test OK
18	UH01-TF02	*UDT_CH_DI_Param...								DUT2 test OK
19	UH01-TF03	*UDT_CH_DI_Param...								SSI test done
20	UH01-TF04	*UDT_CH_DI_Param...								DUT1 write result OK
21	UH01-TF05	*UDT_CH_DI_Param...								DUT2 write result OK
22	UH01-TF06	*UDT_CH_DI_Param...								Write test done
23	UH01-TF07	*UDT_CH_DI_Param...								Watch dog
24	xxxxx_Drive_xxxxx	Byte	16#0							
25	GL01-MA01	*UDT_Motor_Param...								
26	GL01-MA01Drive1	*UDT_Drive1_Para...								
27	xxxxx_Timers_xxxxx	Byte	16#0							
28	GL01-MA01_Timeout	Time	T#6s							
29	xxxxx_Alarms_xxxxx	Byte	16#0							
30	TimeErrGL01MA01_Al...	*UDT_DI_Mon_Para...								Alarm

Obrázek C.2: *DB HMI parametrů stanice 100*

HMI_CMD_100										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	▼ Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	▣ xxxxx__BinarySensors...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	▣ ▸ GL01-BG01	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Turn table stop
4	▣ ▸ GL01-BG02	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Turn table position
5	▣ ▸ UR01-BP01_1	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Pressure sensor out 1
6	▣ ▸ UR01-BP01_2	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Pressure sensor out 2
7	▣ ▣ xxxxx__BinaryOthers_...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
8	▣ ▸ UC01-QA01	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Motor protect for turn table OK
9	▣ ▸ UC01-FC03	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Power supply 5V OK
10	▣ ▸ UC01-FB21	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Light and plugs OK
11	▣ ▸ UC01-FB22	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SSI plugs OK
12	▣ ▸ UH01-FC30	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Protect 24VDC OK
13	▣ ▸ UR01-SF03	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button Start
14	▣ ▸ UR01-SF04	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button reset
15	▣ ▸ UF01-SF01	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button steps
16	▣ ▸ UF01-SF01_13	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button steps is correct
17	▣ ▸ UH01-TF01	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		DUT1 test OK
18	▣ ▸ UH01-TF02	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		DUT2 test OK
19	▣ ▸ UH01-TF03	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SSI test done
20	▣ ▸ UH01-TF04	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		DUT1 write result OK
21	▣ ▸ UH01-TF05	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		DUT2 write result OK
22	▣ ▸ UH01-TF06	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Write test done
23	▣ ▸ UH01-TF07	*UDT_CH_DI_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SSI watch dog
24	▣ ▣ xxxxx__Drive_xxxxx	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
25	▣ ▸ GL01-MA01	*UDT_Motor_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
26	▣ ▸ GL01-MA01Drive1	*UDT_Drive1_Cor...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
27	▣ ▣ xxxxx__Alarms_xxxxx	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
28	▣ ▸ TimeErrGL01MA01_Al...	*UDT_DI_Mon_Co...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
29	▣ ▸ 100UR01-SF01_Alarm	*UDT_DI_Mon_Co...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - emergency button - HMI
30	▣ ▸ 100UR01-SF02_Alarm	*UDT_DI_Mon_Co...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - emergency button - output

Obrázek C.3: DB HMI povelů stanice 100

STATUS_110										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	▼ Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	▣ xxxxx__BinarySensors...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	▣ ▣ BG01	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot A occupied
4	▣ ▣ BG02	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot B occupied
5	▣ xxxxx__BinaryOthers_...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	▣ ▣ UR01-SF01	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Cycle start
7	▣ ▣ xxxxx__MatCard_xxx...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
8	▣ ▣ ▸ Slot_A	*UDT_MatCard*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	▣ ▣ ▸ Slot_B	*UDT_MatCard*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	▣ ▣ Time	Time_Of_Day	TOD#00:00:00	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
11	▣ ▣ xxxxx__Cyclus_xxxxx	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
12	▣ ▣ SlotsActive	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		110 Slots active -> Mat card
13	▣ ▣ SlotsOccupied	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		110 Slots occupied -> Mat card
14	▣ ▣ ReadyToStart	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		110 Ready to start cycle
15	▣ ▣ Cycle_done	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		110 Cycle done - ready for turn
16	▣ ▣ xxxxx__Alarms_xxxxx	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
17	▣ ▣ ▸ 110UR01-KG01R_Alarm	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Alarm - light barrier input

Obrázek C.4: Statusový DB stanice 110

HMI_PARAM_110										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	▼ Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	▣ xxxxx__BinarySensors...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	▣ ▣ UR01-BG01	*UDT_CH_DI_Param...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Slot A occupied
4	▣ ▣ UR01-BG02	*UDT_CH_DI_Param...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Slot B occupied
5	▣ xxxxx__BinaryOthers_...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	▣ ▣ UR01-SF01	*UDT_CH_DI_Param...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button cycle start
7	▣ ▣ xxxxx__SlotA&B_xxx...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
8	▣ ▣ Slot_A_Inactive	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot A is inactive
9	▣ ▣ Slot_B_Inactive	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot B is inactive
10	▣ ▣ xxxxx__Alarms_xxxxx	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
11	▣ ▣ ▸ 110UR01-KG01R_Alarm	*UDT_DI_Mon_Para...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - light barrier input

Obrázek C.5: DB HMI parametrů stanice 110

HMI_CMD_110										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	▼ Static									
2	xxxxx_BinarySensors...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	UR01-BG01	*UDT_CH_DL_Com...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Slot A occupied
4	UR01-BG02	*UDT_CH_DL_Com...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Slot B occupied
5	xxxxx_BinaryOthers_...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	UR01-SF01	*UDT_CH_DL_Com...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button cycle start
7	xxxxx_Alarms_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
8	110UR01-KG01R_Alarm	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - light barrier input

Obrázek C.6: DB HMI povelů stanice 110

STATUS_120										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	▼ Static									
2	xxxxx_BinarySensors...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	BT01-BG01R	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Holding cylindre is back
4	BT01-BG02AR	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Contact cylindre A is back
5	BT01-BG02BR	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Contact cylindre B is back
6	BT01-BG03R	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Mark cylindre A is back
7	BT01-BG04R	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Mark cylindre B is back
8	BT01-BG01V	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Holding cylindre is forward
9	BT01-BG02AV	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Contact cylindre A is forward
10	BT01-BG02BV	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Contact cylindre B is forward
11	BT01-BG03V	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Mark cylindre A is forward
12	BT01-BG04V	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Mark cylindre B is forward
13	xxxxx_BinaryOthers_...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
14	BT01-SF01	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Button request open/close door mess position
15	UR01-QQ01	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Door mess position Lock
16	xxxxx_Drives_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
17	MM01	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Valve holding parts
18	MM02	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Valve contacting A
19	MM03	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Valve mark A
20	MM04	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Valve mark B
21	xxxxx_Door_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
22	120UR01-QQ01_Req...	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Request unlock door
23	120UR01-QQ01_ReqL...	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Request lock door
24	120URR01-QQ01_Clo...	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Door close and lock
25	120UR01-QQ01	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Door mess position
26	xxxxx_MatCard_xxxx...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
27	Slot_A	*UDT_MatCard*			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
28	Slot_B	*UDT_MatCard*			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
29	Time	Time_Of_Day	TOD#00:00:00		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
30	xxxxx_Cyclus_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Obrázek C.7: Statusový DB stanice 120

HMI_PARAM_120										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	▼ Static									
2	xxxxx_BinarySensors...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	BT01-BG01R	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Holding cylindre is back
4	BT01-BG02AR	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contact cylindre A is back
5	BT01-BG02BR	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contact cylindre B is back
6	BT01-BG03R	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Mark cylindre A is back
7	BT01-BG04R	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Mark cylindre B is back
8	BT01-BG01V	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Holding cylindre is forward
9	BT01-BG02AV	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contact cylindre A is forward
10	BT01-BG02BV	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contact cylindre B is forward
11	BT01-BG03V	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Mark cylindre A is forward
12	BT01-BG04V	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Mark cylindre B is forward
13	xxxxx_BinaryOthers_...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
14	BT01-SF01	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button request open/close door mess position
15	UR01-QQ01	*UDT_CH_DL_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Door mess position Lock
16	xxxxx_Drives_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
17	120BT01-MM01	*UDT_Valve2WB_Pa...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Holding valve
18	120BT01-MM02	*UDT_Valve2WB_Pa...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contacting A&B valve
19	120BT01-MM03	*UDT_Valve2WB_Pa...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Marking A valve
20	120BT01-MM04	*UDT_Valve2WB_Pa...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Marking B valve
21	xxxxx_Sequence_xc...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
22	Sequence	*UDT_Sequence_Pa...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
23	xxxxx_Alarms_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
24	120BT01-MM01_Alarm	*UDT_DI_Mon_Para...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
25	120BT01-MM02A_AI...	*UDT_DI_Mon_Para...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
26	120BT01-MM02B_AI...	*UDT_DI_Mon_Para...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
27	120BT01-MM03_Alarm	*UDT_DI_Mon_Para...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
28	120BT01-MM04_Alarm	*UDT_DI_Mon_Para...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
29	120UR01-QQ01_Alarm	*UDT_DI_Mon_Para...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - door mess position

Obrázek C.8: DB HMI parametrů stanice 120

HMI_CMD_120										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	▼ Static									
2	▣ xxxxx_BinarySensors...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	▣ ▶ BT01-BG01R	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Holding cylindre is back
4	▣ ▶ BT01-BG02AR	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contact cylindre A is back
5	▣ ▶ BT01-BG02BR	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contact cylindre B is back
6	▣ ▶ BT01-BG03R	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Mark cylindre A is back
7	▣ ▶ BT01-BG04R	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Mark cylindre B is back
8	▣ ▶ BT01-BG01V	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Holding cylindre is forward
9	▣ ▶ BT01-BG02AV	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contact cylindre A is forward
10	▣ ▶ BT01-BG02BV	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contact cylindre B is forward
11	▣ ▶ BT01-BG03V	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Mark cylindre A is forward
12	▣ ▶ BT01-BG04V	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Mark cylindre B is forward
13	▣ xxxxx_BinaryOthers_...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
14	▣ ▶ UR01-SF01	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button request open/close door mess position
15	▣ ▶ UR01-QQ01	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Door mess position Lock
16	▣ xxxxx_Drives_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
17	▣ ▶ 120BT01-MM01	*UDT_Valve2WB_C...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Holding valve
18	▣ ▶ 120BT01-MM02	*UDT_Valve2WB_C...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contacting A&B valve
19	▣ ▶ 120BT01-MM03	*UDT_Valve2WB_C...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Marking A valve
20	▣ ▶ 120BT01-MM04	*UDT_Valve2WB_C...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Marking B valve
21	▣ xxxxx_Sequence_XX...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
22	▣ ▶ Sequence	*UDT_Sequence_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
23	▣ InitSeq	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
24	▣ xxxxx_Alarms_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
25	▣ ▶ 120BT01-MM01_Alarm	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
26	▣ ▶ 120BT01-MM02A_Al...	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
27	▣ ▶ 120BT01-MM02B_Al...	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
28	▣ ▶ 120BT01-MM03_Alarm	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
29	▣ ▶ 120BT01-MM04_Alarm	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
30	▣ ▶ 120UR01-QQ01_Alarm	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - door mess position

Obrázek C.9: DB HMI povelů stanice 120

STATUS_130										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	▼ Static									
2	▣ xxxxx_BinarySensors...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	▣ CL01-BG01	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1 slot occupied
4	▣ CL01-BG03	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		3 slot occupied
5	▣ CL01-BG05	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		5 slot occupied
6	▣ UR01-BG01	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot A is occupied
7	▣ GM01-BG01	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		End sensor axe X -
8	▣ GM01-BG03R	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Axe Z is up
9	▣ GM01-BP01	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Vakuum is OK
10	▣ CL01-BG02	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		2 slot occupied
11	▣ CL01-BG04	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		4 slot occupied
12	▣ CL01-BG06	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		6 slot occupied
13	▣ UR01-BG02	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot B is occupied
14	▣ GM01-BG02	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		End sensor axe X +
15	▣ GM01-BG03V	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Axe Z is down
16	▣ xxxxx_BinaryOthers_...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
17	▣ CL01-SF01	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Button request open door NOK parts
18	▣ SlotActive1	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1 slot NOK part is active
19	▣ SlotActive2	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		2 slot NOK part is active
20	▣ SlotActive3	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		3 slot NOK part is active
21	▣ SlotActive4	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		4 slot NOK part is active
22	▣ SlotActive5	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		5 slot NOK part is active
23	▣ SlotActive6	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		6 slot NOK part is active
24	▣ xxxxx_Drives_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
25	▣ ▶ TA01	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Axe X
26	▣ ▶ MM01	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Valve axe Z
27	▣ ▶ GS01	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Valve ejector
28	▣ xxxxx_Door_XXXXX	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
29	▣ 130CL01-SF01_RegIn...	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Request input to NOK parts - Stop after cycle

Obrázek C.10: Statusový DB stanice 130



HMI_PARAM_130										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static									
2	xxxxx_BinarySensors...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	CL01-BG01	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		1 slot occupied
4	CL01-BG03	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		3 slot occupied
5	CL01-BG05	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		5 slot occupied
6	UR01-BG01	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Slot A is occupied
7	GM01-BG01	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		End sensor axe X +
8	GM01-BG03R	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Axe Z is up
9	GM01-BP01	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Vakuum is OK
10	CL01-BG02	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		2 slot occupied
11	CL01-BG04	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		4 slot occupied
12	CL01-BG06	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		6 slot occupied
13	UR01-BG02	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Slot B is occupied
14	GM01-BG02	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		End sensor axe X -
15	GM01-BG03V	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Axe Z is down
16	xxxxx_BinaryOthers...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
17	CL01-SF01	*UDT_CH_DI_Param...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button request open door NOK parts
18	xxxxx_Drives_xxxxx	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
19	130GM01-MM01	*UDT_Valve2WB_Pa...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Valve Axe Z
20	130GM01-GS01	*UDT_Valve2WB_Pa...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Ejector
21	xxxxx_Sequence_xx...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
22	Sequence	*UDT_Sequence_Pa...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
23	xxxxx_NumberNOKF...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
24	FollowNOK	Byte	16#3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Number of following NOK parts to alarm
25	xxxxx_NOKSlots_xx...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
26	Slot_inactive_1	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot NOK is inactive
27	Slot_inactive_2	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot NOK is inactive
28	Slot_inactive_3	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot NOK is inactive
29	Slot_inactive_4	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot NOK is inactive
30	Slot_inactive_5	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Slot NOK is inactive

Obrázek C.11: DB HMI parametrů stanice 130

HMI_CMD_130										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static									
2	xxxxx_BinarySensors...	Byte	16#0		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	CL01-BG01	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		1 slot occupied
4	CL01-BG03	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		3 slot occupied
5	CL01-BG05	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		5 slot occupied
6	UR01-BG01	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Slot A is occupied
7	GM01-BG01	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		End sensor axe X +
8	GM01-BG03R	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Axe Z is up
9	GM01-BP01	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Vakuum is OK
10	CL01-BG02	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		2 slot occupied
11	CL01-BG04	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		4 slot occupied
12	CL01-BG06	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		6 slot occupied
13	UR01-BG02	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Slot B is occupied
14	GM01-BG02	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		End sensor axe X -
15	GM01-BG03V	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Axe Z is down
16	xxxxx_BinaryOthers...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
17	CL01-SF01	*UDT_CH_DI_Cor...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Button request open door NOK parts
18	xxxxx_Drives_xxxxx	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
19	130GM01-MM01	*UDT_Valve2WB_C...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Valve Axe Z
20	130GM01-GS01	*UDT_Valve2WB_C...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Ejector
21	130GM01-GS01ManOff	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
22	xxxxx_Sequence_xx...	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
23	Sequence	*UDT_Sequence_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
24	InitSeq	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
25	xxxxx_Alarms_xxxxx	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
26	130GM01-MA01	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - Axe X
27	130GM01-MA01_1	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - Axe X
28	130GM01-MM2	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm
29	130_NOK_Parts	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
30	130UR01-BG03_Alarm	*UDT_DI_Mon_Co...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - Safety sensor NOK parts

Obrázek C.12: DB HMI povelů stanice 130

STATUS_140										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	xxxxx_BinarySensors...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	UR01-BG01	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Detection part slot A - output station
4	UR01-BG02	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Detection part slot B - output station
5	xxxxx_MatCard_xxxx...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	Slot_A	"UDT_MatCard"		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Slot_B	"UDT_MatCard"		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	Time	Time_Of_Day	TOD#00:00:00	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
9	xxxxx_Cyclus_xxxxx	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
10	Cycle_done	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		140 Cycle done - ready for turn
11	xxxxx_Alarms_xxxxx	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
12	140UR01-KG01R_Alarm	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Alarm - light barrier

Obrázek C.13: *Statusový DB stanice 140*

HMI_PARAM_140										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	xxxxx_BinarySensors...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	UR01-BG01	"UDT_CH_DI_Param..."		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Detection part slot A - output station
4	UR01-BG02	"UDT_CH_DI_Param..."		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Detection part slot B - output station
5	xxxxx_Alarms_xxxxx	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	140UR01-KG01R_Alarm	"UDT_DI_Mon_Para..."		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - light barrier input

Obrázek C.14: *DB HMI parametrů stanice 140*

HMI_CMD_140										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	xxxxx_BinarySensors...	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	UR01-BG01	"UDT_CH_DI_Com..."		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Detection part slot A - output station
4	UR01-BG02	"UDT_CH_DI_Com..."		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Detection part slot B - output station
5	xxxxx_Alarms_xxxxx	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	140UR01-KG01R_Alarm	"UDT_DI_Mon_Co..."		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Alarm - light barrier input

Obrázek C.15: *DB HMI povelů stanice 140*

▶	<b>Network 24:</b>	***** Alarm and warning section *****
▶	<b>Network 25:</b>	Alarms - external error
▶	<b>Network 26:</b>	Alarms - step overtime
▶	<b>Network 27:</b>	Alarms - 120BT01-MM01V not at 1
▶	<b>Network 28:</b>	Alarms - 120BT01-MM01V not at 0
▶	<b>Network 29:</b>	Alarms - 120BT01-MM01R not at 1
▶	<b>Network 30:</b>	Alarms - 120BT01-MM01R not at 0
▶	<b>Network 31:</b>	Alarms - 120BT01-MM02V not at 1
▶	<b>Network 32:</b>	Alarms - 120BT01-MM02V not at 0
▶	<b>Network 33:</b>	Alarms - 120BT01-MM02R not at 1
▶	<b>Network 34:</b>	Alarms - 120BT01-MM02R not at 0
▶	<b>Network 35:</b>	Alarms - 120BT01-MM03V not at 1
▶	<b>Network 36:</b>	Alarms - 120BT01-MM03V not at 0
▶	<b>Network 37:</b>	Alarms - 120BT01-MM03R not at 1
▶	<b>Network 38:</b>	Alarms - 120BT01-MM03R not at 0
▶	<b>Network 39:</b>	Alarms - 120BT01-MM04V not at 1
▶	<b>Network 40:</b>	Alarms - 120BT01-MM04V not at 0
▶	<b>Network 41:</b>	Alarms - 120BT01-MM04R not at 1
▶	<b>Network 42:</b>	Alarms - 120BT01-MM04R not at 0
▶	<b>Network 43:</b>	Collective Error

Obrázek C.16: *Sekvenční alarmy stanice 120*

▶	<b>Network 44:</b>	Warning - 120BT01-MM01V not at 1
▶	<b>Network 45:</b>	Warning - 120BT01-MM01V not at 0
▶	<b>Network 46:</b>	Warning - 120BT01-MM01R not at 1
▶	<b>Network 47:</b>	Warning - 120BT01-MM01R not at 0
▶	<b>Network 48:</b>	Warning - 120BT01-MM02V not at 1
▶	<b>Network 49:</b>	Warning - 120BT01-MM02V not at 0
▶	<b>Network 50:</b>	Warning - 120BT01-MM02R not at 1
▶	<b>Network 51:</b>	Warning - 120BT01-MM02R not at 0
▶	<b>Network 52:</b>	Warning - 120BT01-MM03V not at 1
▶	<b>Network 53:</b>	Warning - 120BT01-MM03V not at 0
▶	<b>Network 54:</b>	Warning - 120BT01-MM03R not at 1
▶	<b>Network 55:</b>	Warning - 120BT01-MM03R not at 0
▶	<b>Network 56:</b>	Warning - 120BT01-MM04V not at 1
▶	<b>Network 57:</b>	Warning - 120BT01-MM04V not at 0
▶	<b>Network 58:</b>	Warning - 120BT01-MM04R not at 1
▶	<b>Network 59:</b>	Warning - 120BT01-MM04R not at 0
▶	<b>Network 60:</b>	Warning - 100UH01-TF03 - mess finish
▶	<b>Network 61:</b>	Warning - 120UH01-TF06 - from SSI Write done

Obrázek C.17: *Sekvenční varování stanice 120*

▶	<b>Network 38:</b>	***** Alarm and warning section *****
▶	<b>Network 39:</b>	Alarms - external error
▶	<b>Network 40:</b>	Alarms - step overtime
▶	<b>Network 41:</b>	Alarms - 130BT01-MM01V not at 1
▶	<b>Network 42:</b>	Alarms - 130BT01-MM01V not at 0
▶	<b>Network 43:</b>	Alarms - 130BT01-MM01R not at 1
▶	<b>Network 44:</b>	Alarms - 130BT01-MM01R not at 0
▶	<b>Network 45:</b>	Alarms - 130BT01-GS01V not at 1
▶	<b>Network 46:</b>	Alarms - 130BT01-GS01V not at 0
▶	<b>Network 47:</b>	Alarms - 130BT01-GS01R not at 1
▶	<b>Network 48:</b>	Alarms - 130BT01-GS01R not at 0
▶	<b>Network 49:</b>	Alarms - 130GM01-TA01 home not at 1
▶	<b>Network 50:</b>	Alarms - 130GM01-TA01 slot A not at 1
▶	<b>Network 51:</b>	Alarms - 130GM01-TA01 slot B not at 1
▶	<b>Network 52:</b>	Alarms - 130GM01-TA01 nok not at 1
▶	<b>Network 53:</b>	Alarms - 130GM01-BG01 (End position axe X - ) not at 0
▶	<b>Network 54:</b>	Alarms - 130GM01-BG02 (End position axe X + ) not at 0
▶	<b>Network 55:</b>	Collective Error

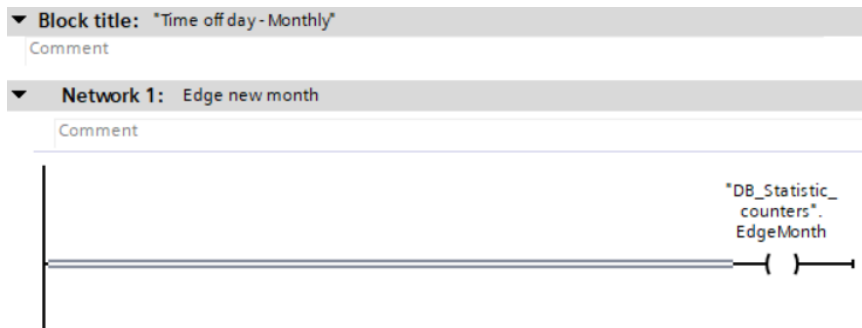
Obrázek C.18: *Sekvenční alarmy stanice 130*

▶	<b>Network 56:</b>	Warning - 130BT01-MM01V not at 1
▶	<b>Network 57:</b>	Warning - 130BT01-MM01V not at 0
▶	<b>Network 58:</b>	Warning - 130BT01-MM01R not at 1
▶	<b>Network 59:</b>	Warning - 130BT01-MM01R not at 0
▶	<b>Network 60:</b>	Warning - 130BT01-GS01V not at 1
▶	<b>Network 61:</b>	Warning - 130BT01-GS01V not at 0
▶	<b>Network 62:</b>	Warning - 130BT01-GS01R not at 1
▶	<b>Network 63:</b>	Warning - 130BT01-GS01R not at 0
▶	<b>Network 64:</b>	Warning - 130GM01-TA01 home not at 1
▶	<b>Network 65:</b>	Warning - 130GM01-TA01 slot A not at 1
▶	<b>Network 66:</b>	Warning - 130GM01-TA01 slot B not at 1
▶	<b>Network 67:</b>	Warning - 130GM01-TA01 nok not at 1

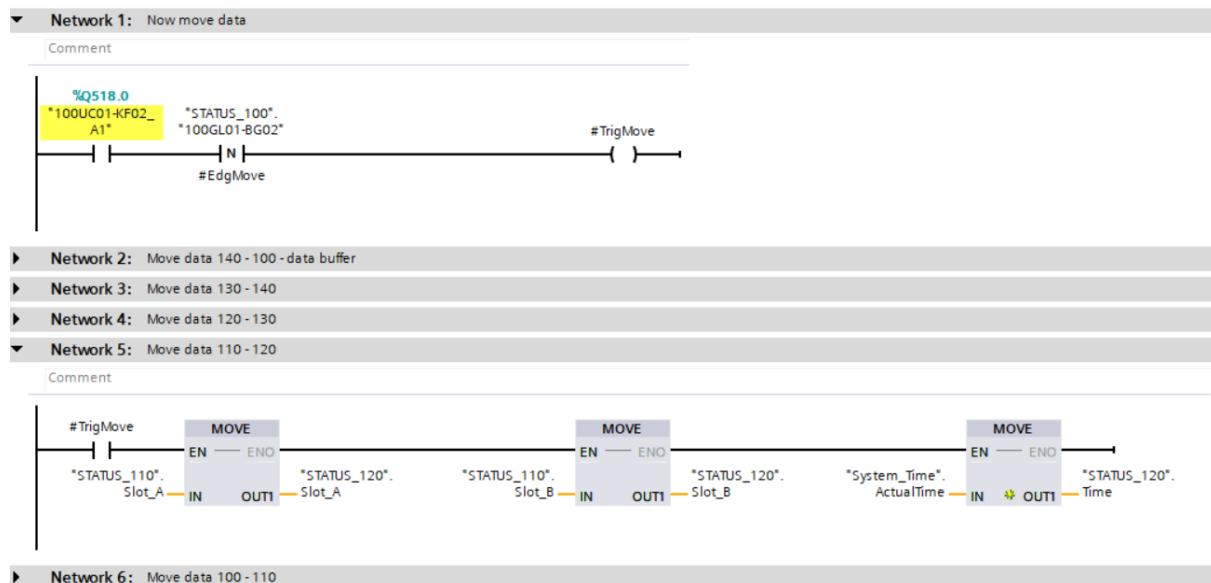
Obrázek C.19: *Sekvenční varování stanice 130*



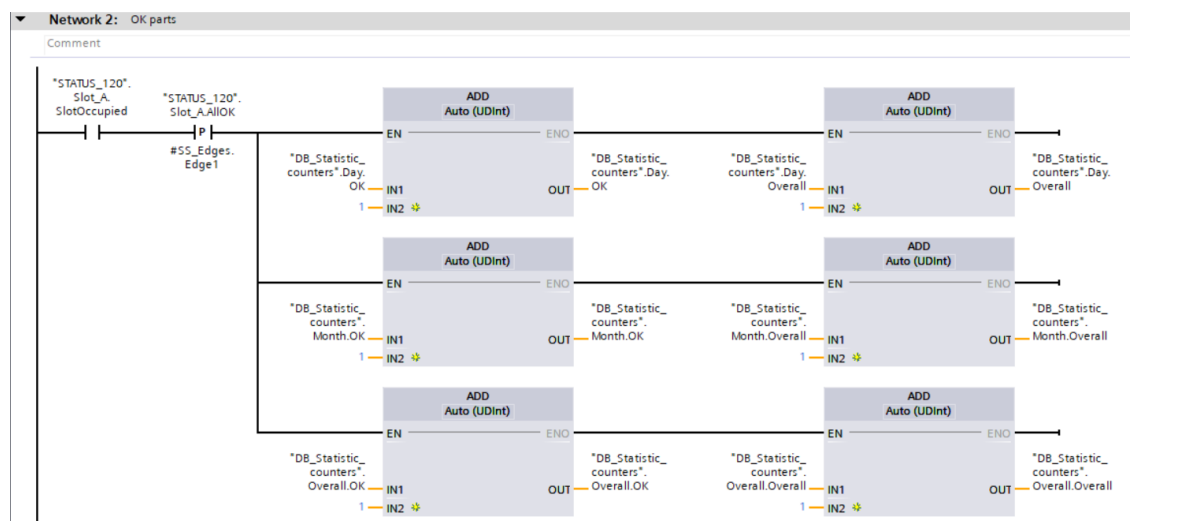
Obrázek C.20: *OB100 – start-up program*



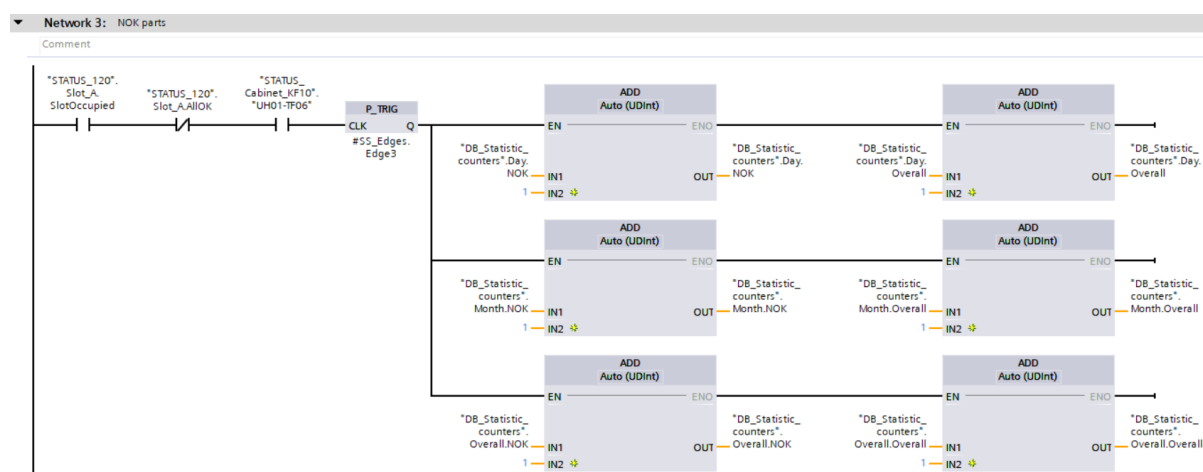
Obrázek C.21: *OB10 –program vykonaný jednou za měsíc*



Obrázek C.22: *FB pro přesun materiálových dat*



Obrázek C.23: *Část FB pro statistická data – materiál OK*



Obrázek C.24: *Část FB pro statistická data – materiál NOK*

**F-runtime group 1 [RTG1]**

**Fail-safe organization block**

Name:

Event class:

Number:

Cycle time:   $\mu$ s

Phase shift:   $\mu$ s

Priority:

**Main safety block**

calls

I-DB:

**F-runtime group parameters**

Warn cycle time of the F-runtime group:   $\mu$ s

Maximum cycle time of the F-runtime group:   $\mu$ s

DB for F-runtime group communication:

F-runtime group information DB:

**Pre/Post processing of the F-runtime group**

Pre processing:

Post processing:

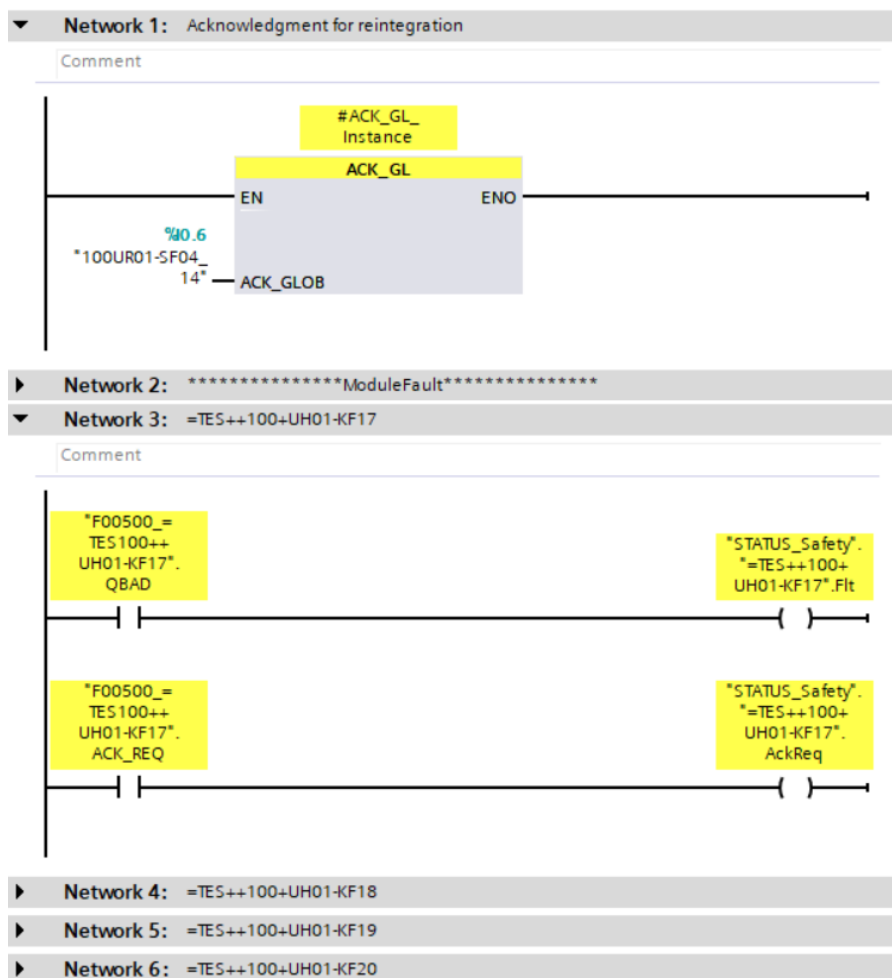
Obrázek C.25: Nastavení bezpečnostního OB

►	Network 1:	*****Interfaces*****
►	Network 2:	Modules
►	Network 3:	Inputs
►	Network 4:	Safe Torque Off - F Interface
►	Network 5:	*****States*****
►	Network 6:	Emergency stop
►	Network 7:	Doors
►	Network 8:	Light barriers
►	Network 9:	*****SafetyZones*****
►	Network 10:	Safety Zone 1 OK
►	Network 11:	Safety Zone 2 OK
►	Network 12:	Safety Zone 3 OK
►	Network 13:	Safety Zone 1 - contactor motor turn table
►	Network 14:	Safety Zone 2 - druck on
►	Network 15:	Safety Zone 3 - linear axe X - STO
►	Network 16:	All safety is OK

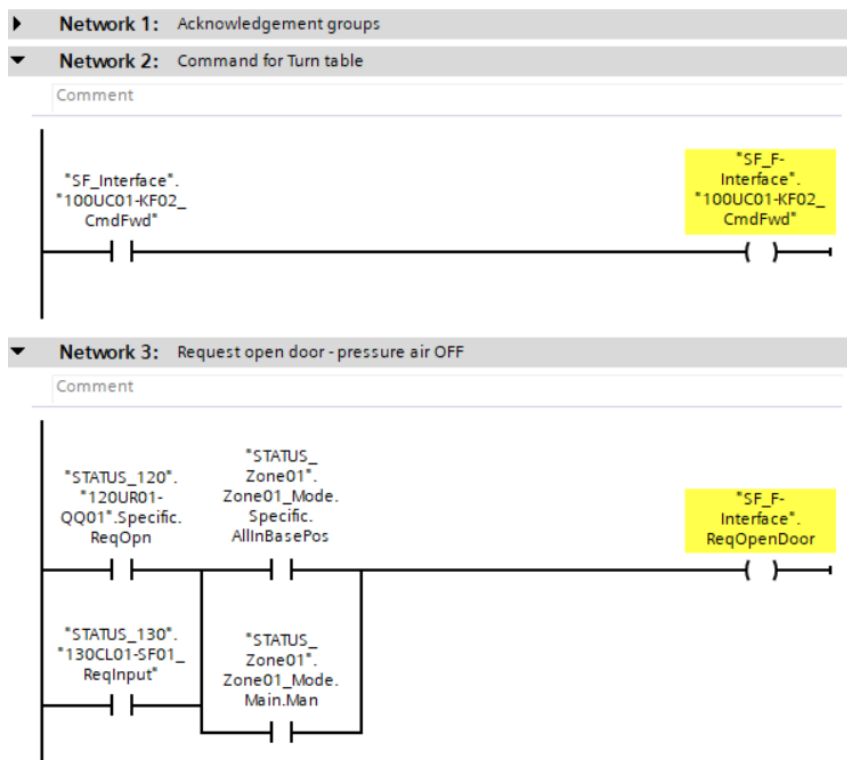
Obrázek C.26: Hlavní FB bezpečnostního programu

▼	Block title: "F-Input"
	Comment
▶	Network 1: *****SafetyZones*****
▶	Network 2: Safety zone 1 - turn table
▶	Network 3: Safety zone 2 - valve terminal
▶	Network 4: Safety zone 3 - linear axe
▶	Network 5: *****EmergencyStopButtons*****
▶	Network 6: 100UR01-SF01 - panel
▶	Network 7: 100UR01-SF02 - output
▶	Network 8: *****Doors*****
▶	Network 9: 120UR01-QQ01 - door testing
▶	Network 10: 130UR01-BG03 - door NOK parts
▶	Network 11: *****LightBarriers*****
▶	Network 12: 110UR01-KG01R - input position
▶	Network 13: 140UR01-KG01R - output position

Obrázek C.27: Funkce zpracovávající bezpečnostní vstupy



Obrázek C.28: Funkce pro reintegraci I/O modulů



Obrázek C.29: *Funkce bezpečnostního programu STO*

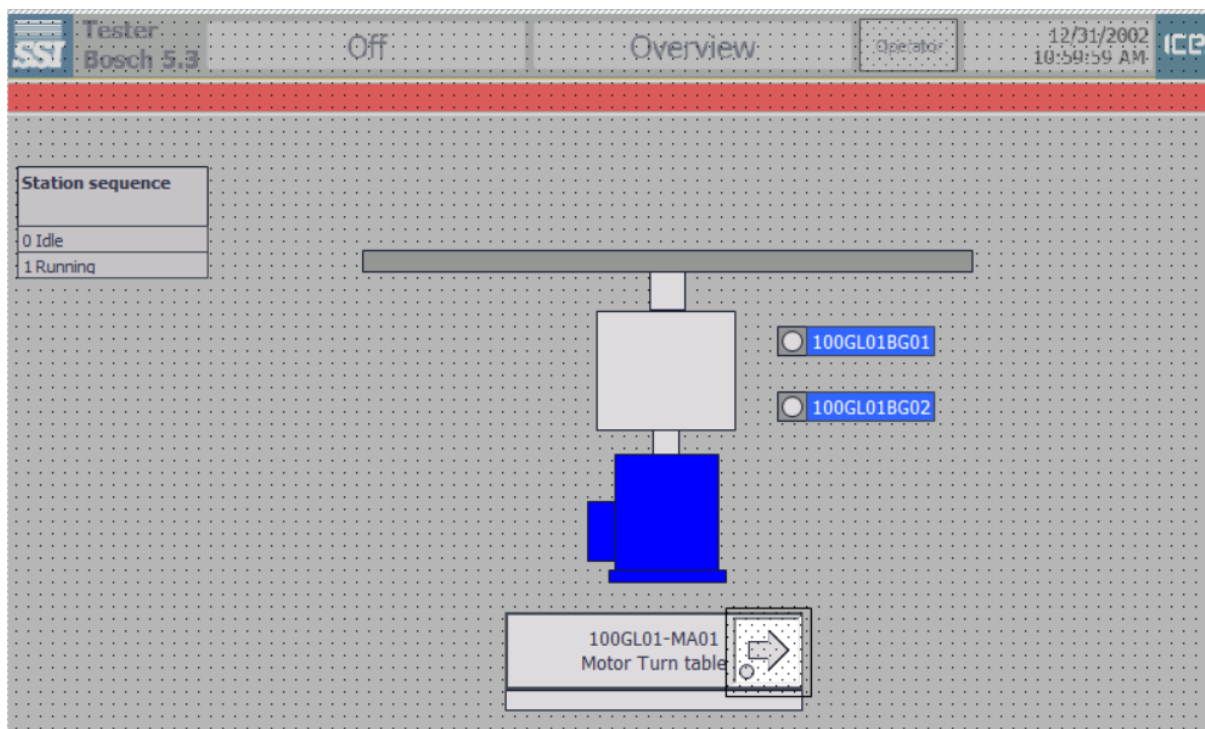
► <b>Network 1:</b>	*****Alarms*****
► <b>Network 2:</b>	*****Zone 1*****
► <b>Network 3:</b>	Zone 1 - all safety zones OK
► <b>Network 4:</b>	Safety OK without lights barriers
► <b>Network 5:</b>	Zone 1 - Error
► <b>Network 6:</b>	Zone 1 - All drives running in automatic
► <b>Network 7:</b>	Zone 1 - There is some drive that can be started
► <b>Network 8:</b>	Zone 1 - Request to suppress alarms from zone
► <b>Network 9:</b>	Zone 1 - Start in automatic is enabled
► <b>Network 10:</b>	Zone 1 - All stations finished the cycle
► <b>Network 11:</b>	Zone 1 - All stations finished the production
► <b>Network 12:</b>	Zone 1 - All stations in base position
► <b>Network 13:</b>	Zone 1 - Main control

Obrázek C.30: *Řízení zóny 1 v hlavním FB zóny 1*

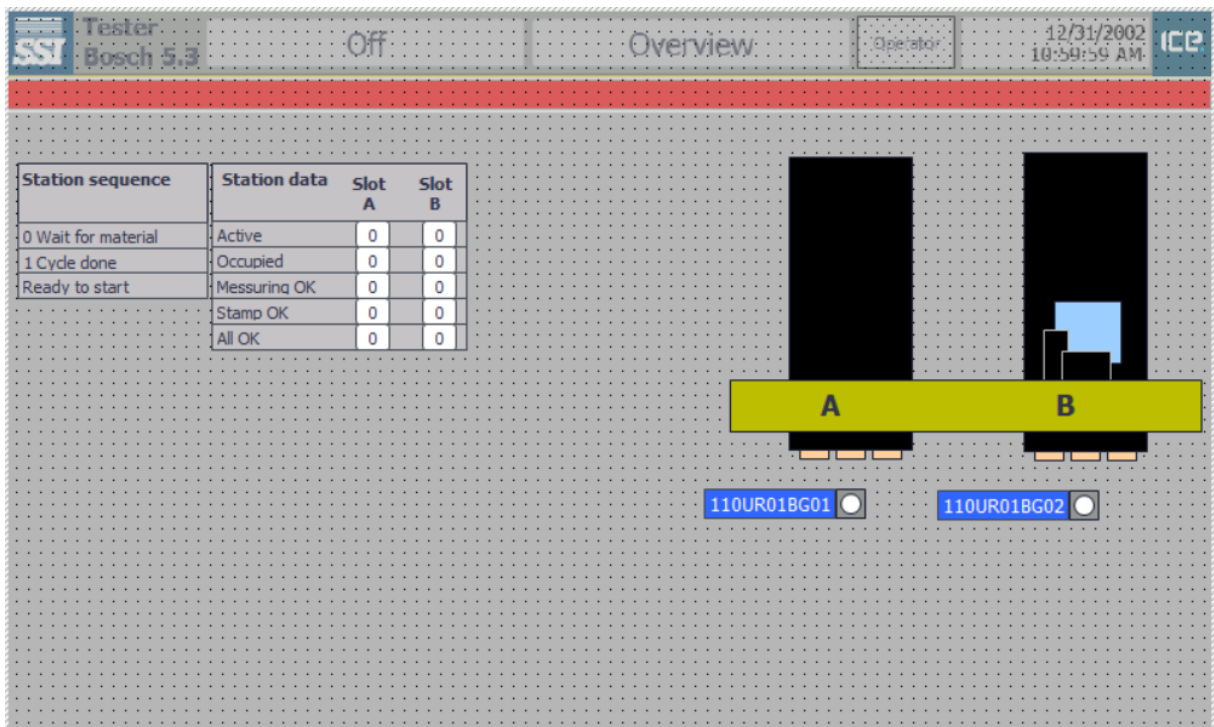


▶ Network 36:	*****Station 120*****
▶ Network 37:	Station 120 - There is some device that is still running
▶ Network 38:	Station 120 - Error to stop
▶ Network 39:	Station 120 - Error (HMI purpose)
▶ Network 40:	Station 120 - All drives running in Automatic
▶ Network 41:	Station 120 - All drives in base position
▶ Network 42:	Station 120 - Single step enable
▶ Network 43:	Station 120 - Single step - request next step
▶ Network 44:	Station 120 - Stop After
▶ Network 45:	Station 120 - Stop Before
▶ Network 46:	Station 120 - End of cycle
▶ Network 47:	Station 120 - End of production
▶ Network 48:	Station 120 - Main control

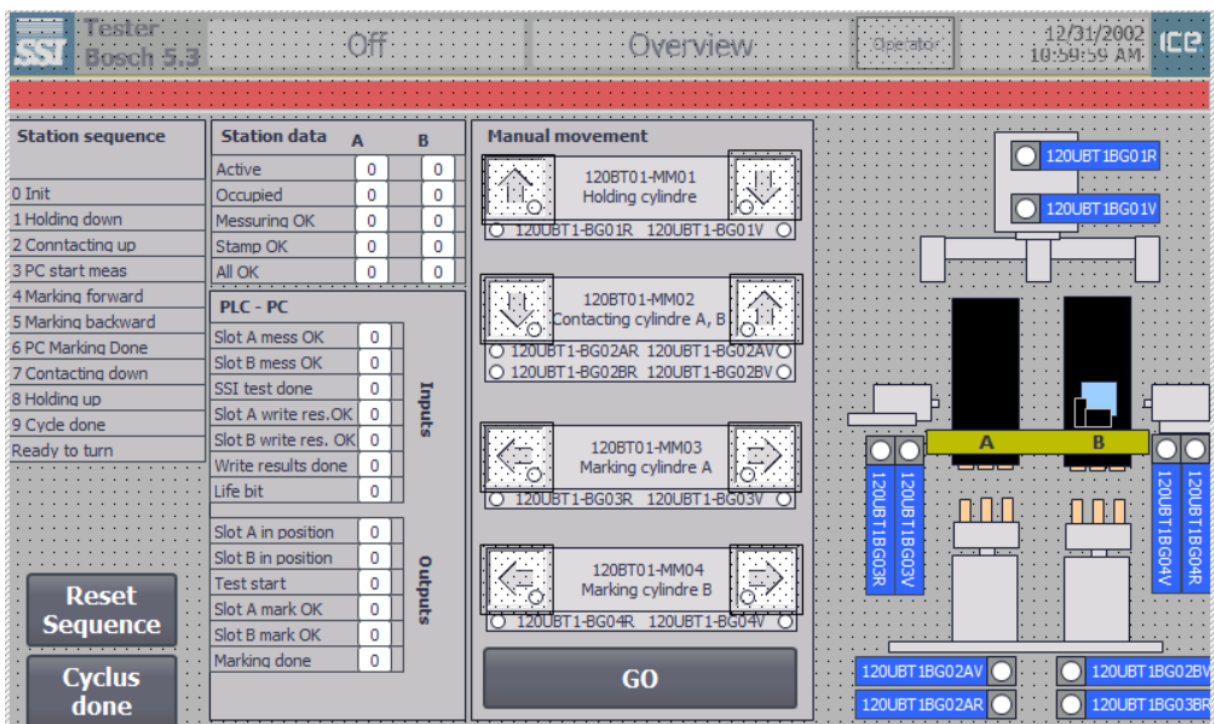
Obrázek C.31: Řízení stanice 120 v hlavním FB zóny 1



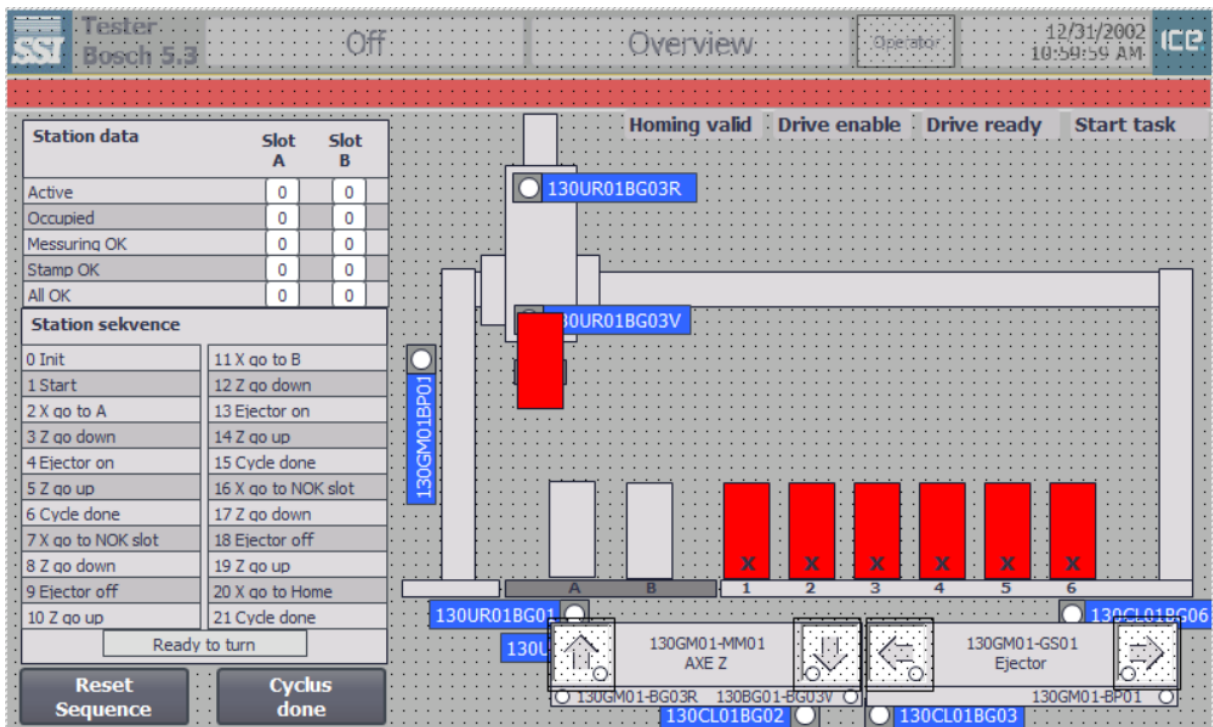
Obrázek C.32: HMI obrazovka stanice 100



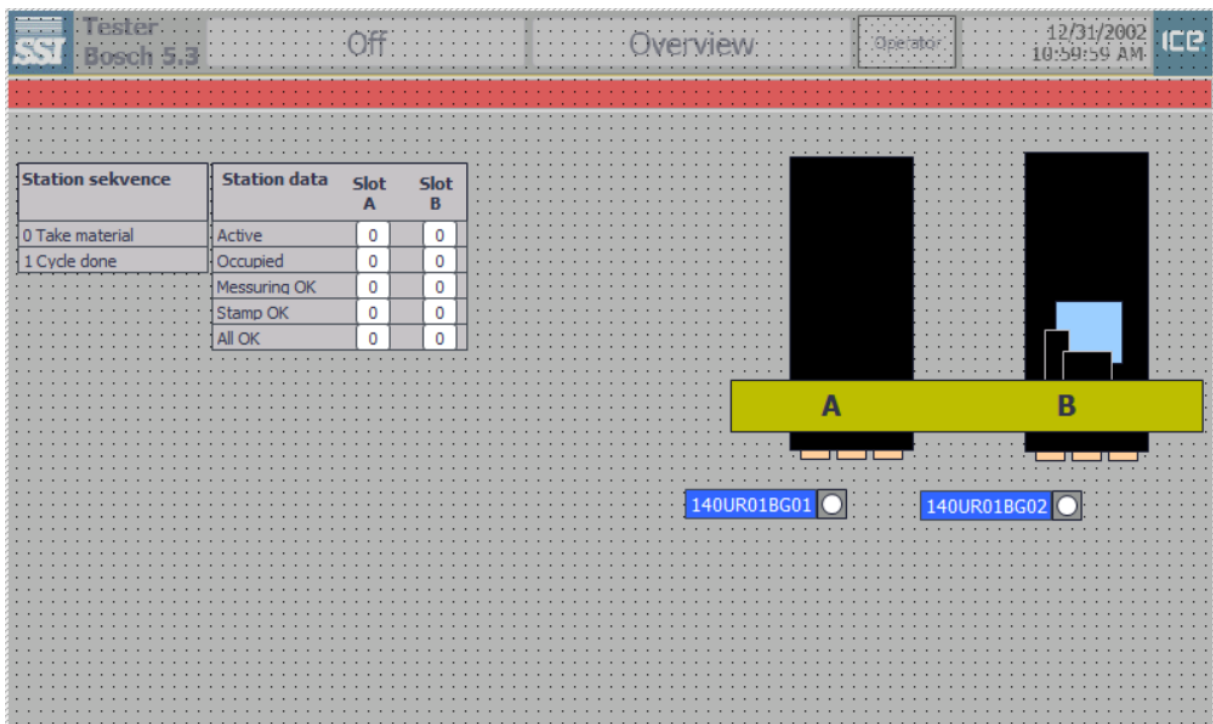
Obrázek C.33: HMI obrazovka stanice 110



Obrázek C.34: HMI obrazovka stanice 120



Obrázek C.35: HMI obrazovka stanice 130



Obrázek C.36: HMI obrazovka stanice 140

<b>Tester</b> SSI Bosch 5.3		Off		Overview		Operator		12/31/2002 10:59:59 AM		CCE																	
<b>Slots settings</b> Slot A active Slot B active		<b>Machine counter</b> OK 0000000000 NOK 0000000000 Total 0000000000		<b>NOK settings</b> Follow 000 Slot active 1 Active Slot active 2 Active Slot active 3 Active Slot active 4 Active Slot active 5 Active Slot active 6 Active		<b>Panel settings</b> 000 [%] Set brightness Transfer Stop runtime Calibrate touch screen Activate clean screen Test light																					
<b>Station 130 - Axe X settings</b> Actual position 0000000000 Velocity [%] 000 Home position Set 0000000000 Slot A Set 0000000000 Slot B Set 0000000000 Position 1 Set 0000000000 Position 2 Set 0000000000 Position 3 Set 0000000000 Position 4 Set 0000000000 Position 5 Set 0000000000 Position 6 Set 0000000000				<b>Setting PLC date and time</b> Set PLC Date and Time 12/31/2002 10:59:59 AM Remote control																							
<b>Administration</b> <table border="1"> <thead> <tr> <th>User</th> <th>Password</th> <th>Group</th> <th>Logoff time</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>								User	Password	Group	Logoff time																
User	Password	Group	Logoff time																								

Obrázek C.36: HMI obrazovka nastavení

Příloha D: *Komunikační aplikace*

```
static void Shm()
{
    using (MemoryMappedFile mmf = MemoryMappedFile.CreateOrOpen("SIMITPLCSHM", headerTotalSizeOfMem))
    {
        bool mutexCreated;
        Mutex mutex = new Mutex(true, "SIMITPLCSHMMutex", out mutexCreated);

        Console.WriteLine("Opening a shared memory Done. ");

        mmfG1 = mmf;
        mutexG1 = mutex;
        Timer myTimer = new Timer(TimerCallback, null, 0, comTime);

        try
        {
            Thread.Sleep(Timeout.Infinite);
        }
        catch (ThreadInterruptedException e)
        {
            Console.WriteLine("SHM is being destroyed . . . ");
            myTimer.Change(Timeout.Infinite, Timeout.Infinite);
        }
    }
    Console.WriteLine("SHM has been destroyed. ");
    comOK = false;
}
```

Obrázek D.1: *Metoda Shm pro vytvoření/otevření prostoru ve sdílené paměti*

```
private static void Shm_data()
{
    try
    {
        mutexG1.WaitOne();
        //Writing to SHM, Data from PLC to SIMIT
        using (MemoryMappedViewStream stream = mmfG1.CreateViewStream())
        {
            BinaryWriter writer = new BinaryWriter(stream);
            writer.Write(headerTotalSizeOfMem);
            writer.Write(headerSizeOfHeader);
            writer.Seek(20, SeekOrigin.Current);
            for (int i = 0; i < plcOutputs.Length; i++)
            {
                writer.Write(plcOutputs[i]);
            }
        }
        //Reading from SHM, Data from SIMIT to PLC
        using (MemoryMappedViewStream stream = mmfG1.CreateViewStream())
        {
            BinaryReader reader = new BinaryReader(stream);
            reader.ReadInt32();
            reader.ReadInt32();
            for (int i = 0; i < plcInputs.Length; i++)
            {
                plcInputs[i] = reader.ReadByte();
            }
        }
        mutexG1.ReleaseMutex();
    }
    catch (Exception e)
    {
        if (abortShm == false)
        {
            Console.WriteLine("SHM data exchange problem. ");
            Console.WriteLine("{0} ", e);
        }
        comOK = false;
    }
}
```

Obrázek D.2: *Metoda Shm\_data pro zápis a čtení dat do/ze sdílené paměti*

```

static void Plc(string PLCName)
{
    try
    {
        //Used for local instance
        IInstance myInstance_ = SimulationRuntimeManager.CreateInterface(PLCName);

        //Used for remote instance
        //IRemoteRuntimeManager myRemoteInstance = SimulationRuntimeManager.RemoteConnect("192.168.1.101:50000");
        //IInstance myInstance = myRemoteInstance.CreateInterface("1515F");

        Console.WriteLine("Connecting to the PLCSIM instance Done. ");

        myInstance_.IsAlwaysSendOnEndOfCycleEnabled = true;
        myInstance_.CommunicationInterface = ECommunicationInterface.Softbus;
        myInstance = myInstance_;
        myInstance_.OnConfigurationChanged += myInstance_OnConfigurationChanged;
        UpdateTags();
        myInstance_.OnEndOfCycle += myInstance_OnEndOfCycle;

        comOK = true;
    }
    catch (Exception e)
    {
        Console.WriteLine("PLCSIM Adv instance was not found. ");
        comOK = false;
    }
}

```

Obrázek D.3: *Metoda Plc pro připojení k instanci PLC*

```

static void ReadwriteIO()
{
    //Console.WriteLine("Writing Input signals to PLCSIM...");
    //Vstupy PLC
    //byte 0
    ushort i = 0;
    myInstance.WriteBool("100UC01-QA01_14", ConvertByteToBool(plcInputs[i], 0));
    myInstance.WriteBool("100UC01-FC03_14", ConvertByteToBool(plcInputs[i], 1));
    myInstance.WriteBool("100UC01-FB21_14", ConvertByteToBool(plcInputs[i], 2));
    myInstance.WriteBool("100UC01-FB22_14", ConvertByteToBool(plcInputs[i], 3));
    myInstance.WriteBool("100UH01-FC30_14", ConvertByteToBool(plcInputs[i], 4));
    myInstance.WriteBool("100UR01-SF03_14", ConvertByteToBool(plcInputs[i], 5));
    myInstance.WriteBool("100UR01-SF04_14", ConvertByteToBool(plcInputs[i], 6));
    myInstance.WriteBool("100UF01-SF01_14", ConvertByteToBool(plcInputs[i], 7));
    //byte 1
    i = 1;
    myInstance.WriteBool("100UF01-SF01_13", ConvertByteToBool(plcInputs[i], 0));
    myInstance.WriteBool("110UR01-SF01_14", ConvertByteToBool(plcInputs[i], 1));
    myInstance.WriteBool("120BT01-SF01_14", ConvertByteToBool(plcInputs[i], 2));
    myInstance.WriteBool("130CL01-SF01_14", ConvertByteToBool(plcInputs[i], 3));
}

```

Obrázek D.4: *Část metody ReadWriteIO - zápis vstupů PLC pomocí API*

```

//Console.WriteLine("Reading Output signals from PLCSIM...");
//Vystupy PLC
bool[] b = new bool[8];
//byte 0
i = 0;
b[0] = myInstance.ReadBool("100UR01-PF03_X1");
b[1] = myInstance.ReadBool("100UR01-PF04_X1");
b[2] = myInstance.ReadBool("110UR01-PF01_X1");
b[3] = myInstance.ReadBool("110UR01-PF02_X1");
b[4] = myInstance.ReadBool("120BT01-PF01_X1");
b[5] = myInstance.ReadBool("130CL01-PF01_X1");
b[6] = myInstance.ReadBool("140UR01-PF01_X1");
b[7] = myInstance.ReadBool("120UR01-QQ01_E1");
plcOutputs[i] = Encodebool(b);
//byte 1
i = 1;
b[0] = myInstance.ReadBool("RemoteControlON");
b[1] = false; //myInstance.ReadBool("100UR01-PF04_X1");
b[2] = false; //myInstance.ReadBool("110UR01-PF01_X1");
b[3] = false; //myInstance.ReadBool("110UR01-PF02_X1");
b[4] = false; //myInstance.ReadBool("120BT01-PF01_X1");
b[5] = false; //myInstance.ReadBool("130CL01-PF01_X1");
b[6] = false; //myInstance.ReadBool("140UR01-PF01_X1");
b[7] = false; //myInstance.ReadBool("120UR01-QQ01_E1");
plcOutputs[i] = Encodebool(b);

```

Obrázek D.5: Část metody *ReadWriteIO*, čtení výstupů PLC pomocí API

Tabulka D.6: Datová oblast ve sdílené paměti pro PLC vstupy

Symbolická adresa	Datový typ	Fyzická adresa	Komentář	SHM adresa
100UC01-QA01_14	Bool	%I0.0	Protection of turn table motor is OK	%M0.0
100UC01-FC03_14	Bool	%I0.1	Power supply 5VDC OK	%M0.1
100UC01-FB21_14	Bool	%I0.2	Light and plugs 230V OK	%M0.2
100UC01-FB22_14	Bool	%I0.3	SSI plugs 230V OK	%M0.3
100UH01-FC30_14	Bool	%I0.4	Circuit breaker 24VDC OK	%M0.4
100UR01-SF03_14	Bool	%I0.5	Button start	%M0.5
100UR01-SF04_14	Bool	%I0.6	Button reset	%M0.6
100UF01-SF01_14	Bool	%I0.7	Button step	%M0.7
100UF01-SF01_13	Bool	%I1.0	Button step - connect	%M1.0
110UR01-SF01_14	Bool	%I1.1	Buuton cycle start	%M1.1
120BT01-SF01_14	Bool	%I1.2	Button request open/close door mess position	%M1.2
130CL01-SF01_14	Bool	%I1.3	Button request open door NOK parts	%M1.3
120UR01-QQ01_11	Bool	%I1.4	Door mess position Lock	%M1.4
I1.5	Bool	%I1.5	Reserve	%M1.5
I1.6	Bool	%I1.6	Reserve	%M1.6
I1.7	Bool	%I1.7	Reserve	%M1.7
100UH01-TF01_14	Bool	%I2.0	DUT1 test OK	%M2.0
100UH01-TF02_14	Bool	%I2.1	DUT2 test OK	%M2.1
100UH01-TF03_14	Bool	%I2.2	SSI test done	%M2.2
100UH01-TF04_14	Bool	%I2.3	DUT1 write result OK	%M2.3
100UH01-TF05_14	Bool	%I2.4	DUT2 write result OK	%M2.4
100UH01-TF06_14	Bool	%I2.5	Write test done	%M2.5
100UH01-TF07_14	Bool	%I2.6	SSI tester ready	%M2.6

I2.7	Bool	%I2.7	Reserve	%M2.7
I3.0	Bool	%I3.0	Reserve	%M3.0
I3.1	Bool	%I3.1	Reserve	%M3.1
I3.2	Bool	%I3.2	Reserve	%M3.2
I3.3	Bool	%I3.3	Reserve	%M3.3
I3.4	Bool	%I3.4	Reserve	%M3.4
I3.5	Bool	%I3.5	Reserve	%M3.5
I3.6	Bool	%I3.6	Reserve	%M3.6
I3.7	Bool	%I3.7	Reserve	%M3.7
I10UR01-BG01_4	Bool	%I108.0	Detection part slot A - input station	%M4.0
I40UR01-BG01_4	Bool	%I108.1	Detection part slot A - output station	%M4.1
I00GL01-BG02	Bool	%I108.2	Sensor turn table in position	%M4.2
I108.3	Bool	%I108.3	Reserve	%M4.3
I108.4	Bool	%I108.4	Reserve	%M4.4
I108.5	Bool	%I108.5	Reserve	%M4.5
I108.6	Bool	%I108.6	Reserve	%M4.6
I108.7	Bool	%I108.7	Reserve	%M4.7
I10UR01-BG02_4	Bool	%I109.0	Detection part slot B - input station	%M5.0
I40UR01-BG02_4	Bool	%I109.1	Detection part slot B - output station	%M5.1
I00GL01-BG01	Bool	%I109.2	Sensor stop - turn table	%M5.2
I109.3	Bool	%I109.3	Reserve	%M5.3
I109.4	Bool	%I109.4	Reserve	%M5.4
I109.5	Bool	%I109.5	Reserve	%M5.5
I109.6	Bool	%I109.6	Reserve	%M5.6
I109.7	Bool	%I109.7	Reserve	%M5.7
I20BT01-BG01R_4	Bool	%I110.0	Holding cylindre is back	%M6.0
I20BT01-BG02AR_4	Bool	%I110.1	Contact cylindre A is back	%M6.1
I20BT01-BG02BR_4	Bool	%I110.2	Contact cylindre B is back	%M6.2
I20BT01-BG03R_4	Bool	%I110.3	Mark cylindre A is back	%M6.3
I20BT01-BG04R_4	Bool	%I110.4	Mark cylindre B is back	%M6.4
I110.5	Bool	%I110.5	Reserve	%M6.5
I110.6	Bool	%I110.6	Reserve	%M6.6
I110.7	Bool	%I110.7	Reserve	%M6.7
I20BT01-BG01V_4	Bool	%I111.0	Holding cylindre is forward	%M7.0
I20BT01-BG02AV_4	Bool	%I111.1	Contact cylindre A is forward	%M7.1
I20BT01-BG02BV_4	Bool	%I111.2	Contact cylindre B is forward	%M7.2
I20BT01-BG03V_4	Bool	%I111.3	Mark cylindre A is forward	%M7.3
I20BT01-BG04V_4	Bool	%I111.4	Mark cylindre B is forward	%M7.4
I111.5	Bool	%I111.5	Reserve	%M7.5
I111.6	Bool	%I111.6	Reserve	%M7.6
I111.7	Bool	%I111.7	Reserve	%M7.7
I30CL01-BG01_4	Bool	%I116.0	1 slot occupied	%M8.0
I30CL01-BG03_4	Bool	%I116.1	3 slot occupied	%M8.1



130CL01-BG05_4	Bool	%I116.2	5 slot occupied	%M8.2
130UR01-BG01_4	Bool	%I116.3	Slot A is occupied	%M8.3
130GM01-BG01_4	Bool	%I116.4	End sensor axe X +	%M8.4
130GM01-BG03R	Bool	%I116.5	Axe Z is up	%M8.5
130GM01-BP01_4	Bool	%I116.6	Vakuum is OK	%M8.6
I116.7	Bool	%I116.7	Reserve	%M8.7
130CL01-BG02_4	Bool	%I117.0	2 slot occupied	%M9.0
130CL01-BG04_4	Bool	%I117.1	4 slot occupied	%M9.1
130CL01-BG06_4	Bool	%I117.2	6 slot occupied	%M9.2
130UR01-BG02_4	Bool	%I117.3	Slot B is occupied	%M9.3
130GM01-BG02_4	Bool	%I117.4	End sensor axe X -	%M9.4
130GM01-BG03V_4	Bool	%I117.5	Axe Z is down	%M9.5
I117.6	Bool	%I117.6	Reserve	%M9.6
I117.7	Bool	%I117.7	Reserve	%M9.7
100UR01-BP01_1	Bool	%I149.0	Pressure sensor out 1	%M10.0
100UR01-BP01_2(1)	Bool	%I149.1	Pressure sensor out 2	%M10.1
100UR01-SF01_12	Bool	%I500.0	Emergency stop button - panel	%M10.2
100UR01-SF02_12	Bool	%I500.1	Emergency stop button - output	%M10.3
120UR01-QQ01_21	Bool	%I500.2	Door testing position - close and lock	%M10.4
130UR01-BG03_4	Bool	%I500.3	Sensor door NOK parts - close	%M10.5
110UR01-KG01R_4	Bool	%I506.0	Light barrier input position	%M10.6
140UR01-KG01R_4	Bool	%I506.1	Light barrier output position	%M10.7
100UC01-KF02_21	Bool	%I512.0	Feedback - Turn table	%M11.0
100UH01-KF30_Y2	Bool	%I512.1	Feedback - Valve terminal	%M11.1
100UR01-BP01_2	Bool	%I512.2	Pressure air is 0	%M11.2
100UC01-TA01_EDM	Bool	%I512.3	Feedback - Linear axe X	%M11.3
CMMO_IN_1	Byte	%IB132	SCON	%MB12
CMMO_IN_2	Byte	%IB133	SPOS	%MB13
CMMO_IN_3	Byte	%IB134	SDIR	%MB14
CMMO_IN_4	Byte	%IB135	Actual value 1	%MB15
CMMO_IN_5678	DInt	%ID136	Actual value 2	%MD16

Tabulka D.7: *Datová oblast ve sdílené paměti pro PLC výstupy*

Symbolická adresa	Datový typ	Fyzická adresa	Komentář	SHM adresa
100UR01-PF03_X1	Bool	%Q0.0	Light Start	%M20.0
100UR01-PF04_X1	Bool	%Q0.1	Light Reset	%M20.1
110UR01-PF01_X1	Bool	%Q0.2	Light Cycle start color 1	%M20.2
110UR01-PF02_X1	Bool	%Q0.3	Light Cycle start color 2	%M20.3
120BT01-PF01_X1	Bool	%Q0.4	Light button request door open/close mess position	%M20.4
130CL01-PF01_X1	Bool	%Q0.5	Light button request open door NOK parts	%M20.5
140UR01-PF01_X1	Bool	%Q0.6	Light output position - possibility	%M20.6
120UR01-QQ01_E1	Bool	%Q0.7	Door mess position lock	%M20.7

RemoteControlON	Bool	%Q1.0	Remote control ON	%M21.0
Q1.1	Bool	%Q1.1	Reserve	%M21.1
Q1.2	Bool	%Q1.2	Reserve	%M21.2
Q1.3	Bool	%Q1.3	Reserve	%M21.3
Q1.4	Bool	%Q1.4	Reserve	%M21.4
Q1.5	Bool	%Q1.5	Reserve	%M21.5
Q1.6	Bool	%Q1.6	Reserve	%M21.6
Q1.7	Bool	%Q1.7	Reserve	%M21.7
100UH01-TF11_A1	Bool	%Q2.0	DUT1 in position	%M22.0
100UH01-TF12_A1	Bool	%Q2.1	DUT2 in position	%M22.1
100UH01-TF13_A1	Bool	%Q2.2	Test start	%M22.2
100UH01-TF14_A1	Bool	%Q2.3	DUT1 mark OK	%M22.3
100UH01-TF15_A1	Bool	%Q2.4	DUT2 mark OK	%M22.4
100UH01-TF16_A1	Bool	%Q2.5	Marking done	%M22.5
Q2.6	Bool	%Q2.6	Reserve	%M22.6
Q2.7	Bool	%Q2.7	Reserve	%M22.7
Q3.0	Bool	%Q3.0	Reserve	%M23.0
Q3.1	Bool	%Q3.1	Reserve	%M23.1
Q3.2	Bool	%Q3.2	Reserve	%M23.2
Q3.3	Bool	%Q3.3	Reserve	%M23.3
Q3.4	Bool	%Q3.4	Reserve	%M23.4
Q3.5	Bool	%Q3.5	Reserve	%M23.5
Q3.6	Bool	%Q3.6	Reserve	%M23.6
Q3.7	Bool	%Q3.7	Reserve	%M23.7
120BT01-MM01-R_14	Bool	%Q100.0	Holding cylindre command Up	%M24.0
120BT01-MM01-V_12	Bool	%Q100.1	Holding cylindre command Down	%M24.1
120BT01-MM02-R_14	Bool	%Q100.2	Contact cylindre command Down	%M24.2
120BT01-MM02-V_12	Bool	%Q100.3	Contact cylindre command Up	%M24.3
120BT01-MM03-R_14	Bool	%Q100.4	Mark cylindre A command Back	%M24.4
120BT01-MM03-V_12	Bool	%Q100.5	Mark cylindre A command Forward	%M24.5
120BT01-MM04-R_14	Bool	%Q100.6	Mark cylindre B command Back	%M24.6
120BT01-MM04-V_12	Bool	%Q100.7	Mark cylindre B command Forward	%M24.7
130GM01-MM01-R_14	Bool	%Q101.0	Axe Z command Up	%M25.0
130GM01-MM02-V_12	Bool	%Q101.1	Axe Z command Down	%M25.1
Q101.2	Bool	%Q101.2	Reserve	%M25.2
130GM01-GS01-V_12	Bool	%Q101.3	Ejector command on	%M25.3
Q101.4	Bool	%Q101.4	Reserve	%M25.4
Q101.5	Bool	%Q101.5	Reserve	%M25.5
Q101.6	Bool	%Q101.6	Reserve	%M25.6
Q101.7	Bool	%Q101.7	Reserve	%M25.7
VFPP01_Seg1_green	Bool	%Q104.0	Beacon segment 1 green	%M26.0
VFPP01_Seg1_Red	Bool	%Q104.1	Beacon segment 1 red	%M26.1
VFPP01_Seg1_Blue	Bool	%Q104.2	Beacon segment 1blue	%M26.2

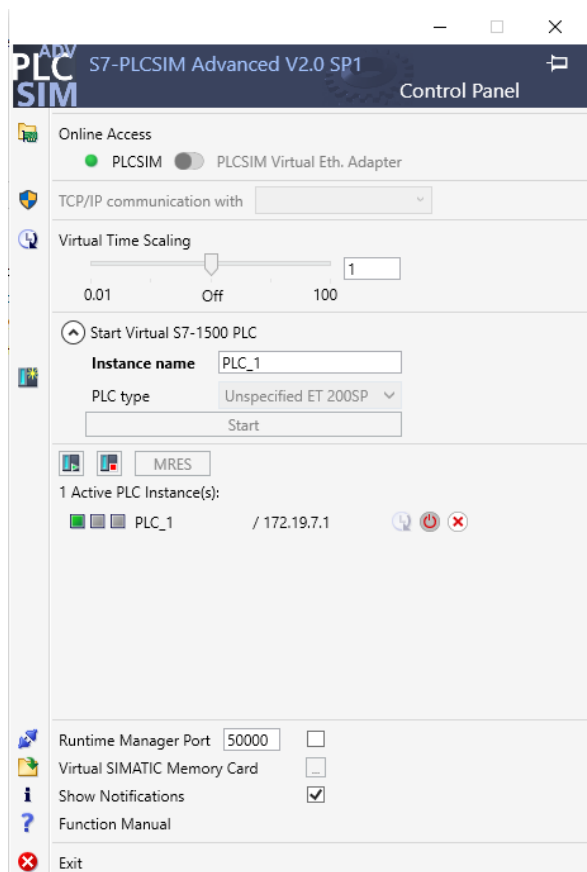
VFPPF01_Seg1_blink	Bool	%Q104.3	Beacon segment 1 blink	%M26.3
VFPPF01_Seg2_Green	Bool	%Q104.4	Beacon segment 2 green	%M26.4
VFPPF01_Seg2_Red	Bool	%Q104.5	Beacon segment 2 red	%M26.5
VFPPF01_Seg2_Blue	Bool	%Q104.6	Beacon segment 2 blue	%M26.6
VFPPF01_Seg2_Blink	Bool	%Q104.7	Beacon segment 2 blink	%M26.7
VFPPF01_Seg3_Green	Bool	%Q105.0	Beacon segment 3 green	%M27.0
VFPPF01_Seg3_Red	Bool	%Q105.1	Beacon segment 3 red	%M27.1
VFPPF01_Seg3_Blue	Bool	%Q105.2	Beacon segment 3 blue	%M27.2
VFPPF01_Seg3_Blink	Bool	%Q105.3	Beacon segment 3 blink	%M27.3
Tag_10	Bool	%Q105.4	Reserve	%M27.4
VFPPF01_Horn	Bool	%Q106.7	Beacon horn	%M27.5
CMMO_OUT_1	Byte	%QB132	CCON	%MB29
CMMO_OUT_2	Byte	%QB133	CPOS	%MB30
CMMO_OUT_3	Byte	%QB134	CDIR	%MB31
CMMO_OUT_4	Byte	%QB135	Setpoint value 1	%MB32
CMMO_OUT_5678	DInt	%QD136	Setpoint value 2	%MD33

MCDSHM (SHM)				
<div> <div> <div></div> <div></div> <div></div> </div> <div>Reset filter</div> </div>				
▼ Inputs				
	Symbol name	Address	Data type	Comment
	<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	36 II_100_Majak_Barva	MW0	INT	
<div></div>	36 II_110_PotvrzovacíTlactko_Barva	MW2	INT	
<div></div>	IB_100_Majak_En	M4.0	BOOL	
<div></div>	IB_110_PotvrzovacíTlactko_En	M4.1	BOOL	
<div></div>	IB_AddMaterial_Slot1A	M4.2	BOOL	
<div></div>	IB_AddMaterial_Slot1B	M4.3	BOOL	
<div></div>	IB_TakeMaterial_Slot4A	M4.4	BOOL	
<div></div>	IB_TakeMaterial_Slot4B	M4.5	BOOL	
<div></div>	IB_TakeMaterial_Slot1A	M4.6	BOOL	
<div></div>	IB_TakeMaterial_Slot1B	M4.7	BOOL	
<div></div>	0 II_120_KK01_Pozice	MW5	INT	
<div></div>	0 II_120_KK02A_Pozice	MW7	INT	
▼ Outputs				
	Symbol name	Address	Data type	Comment
	<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	OB_110_BG01	M20.0	BOOL	
<div></div>	OB_110_BG02	M20.1	BOOL	
<div></div>	OB_130_BG01	M20.2	BOOL	
<div></div>	OB_130_BG02	M20.3	BOOL	
<div></div>	OB_140_BG01	M20.4	BOOL	
<div></div>	OB_140_BG02	M20.5	BOOL	
<div></div>	OB_130_BP01	M20.6	BOOL	
<div></div>	OB_130_CL01_BG01	M20.7	BOOL	
<div></div>	OB_130_CL01_BG02	M23.0	BOOL	
<div></div>	OB_130_CL01_BG03	M23.1	BOOL	
<div></div>	OB_130_CL01_BG04	M23.2	BOOL	
<div></div>	OB_130_CL01_BG05	M23.3	BOOL	

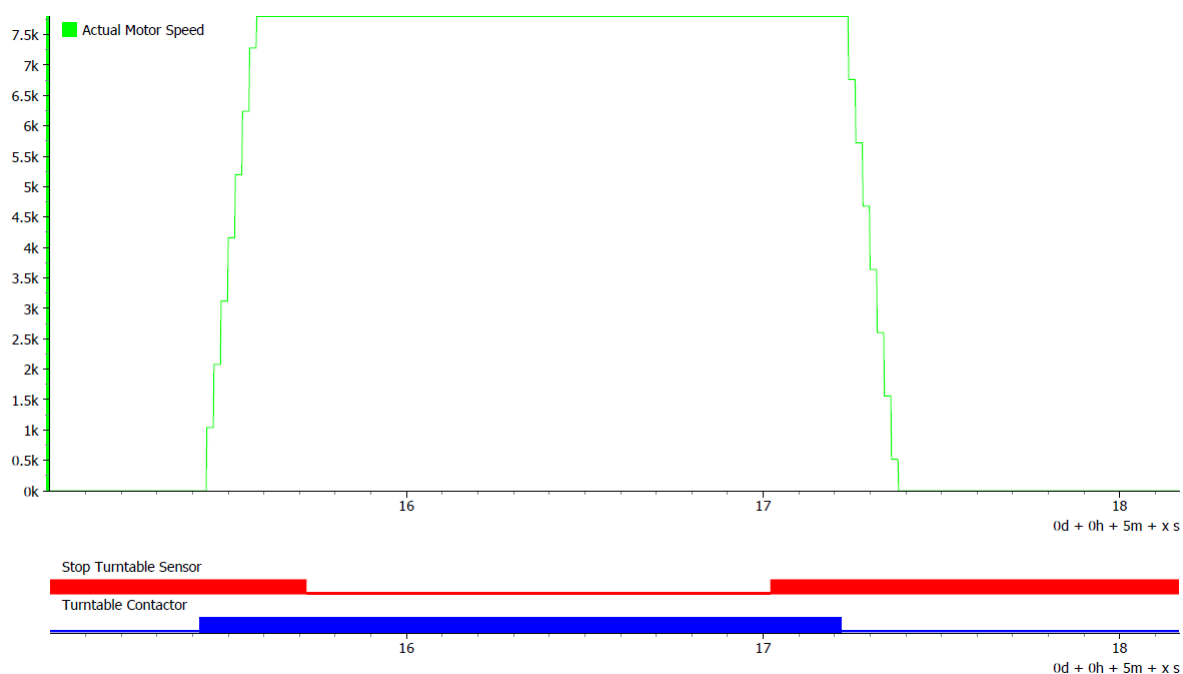
Obrázek E.1: Výměna signálů mezi nástroji NX MCD a SIMIT zobrazena v SIMIT

PLCSHM (SHM)				
<div> <div></div> <div></div> <div></div> </div>				
▼ Inputs <span>Reset filter</span>				
	Symbol name	Address	Data type	Comment
<input checked="" type="checkbox"/>	100UC01-QA01_14	M0.0	BOOL	Protection of turn table motor is OK
<input checked="" type="checkbox"/>	100UC01-FC03_14	M0.1	BOOL	Power supply 5VDC OK
<input checked="" type="checkbox"/>	100UC01-FB21_14	M0.2	BOOL	Light and plugs 230V OK
<input checked="" type="checkbox"/>	100UC01-FB22_14	M0.3	BOOL	SSI plugs 230V OK
<input checked="" type="checkbox"/>	100UH01-FC30_14	M0.4	BOOL	Circuit breaker 24VDC OK
<input checked="" type="checkbox"/>	100UR01-SF03_14	M0.5	BOOL	Button start
<input checked="" type="checkbox"/>	100UR01-SF04_14	M0.6	BOOL	Button reset
<input checked="" type="checkbox"/>	100UF01-SF01_14	M0.7	BOOL	Button step
<input checked="" type="checkbox"/>	100UF01-SF01_13	M1.0	BOOL	Button step - connect
<input checked="" type="checkbox"/>	110UR01-SF01_14	M1.1	BOOL	Button cycle start
<input checked="" type="checkbox"/>	120BT01-SF01_14	M1.2	BOOL	Button request open/close door mess position
<input checked="" type="checkbox"/>	130CL01-SF01_14	M1.3	BOOL	Button request open door NOK parts
▼ Outputs <span>Reset filter</span>				
	Symbol name	Address	Data type	Comment
<input checked="" type="checkbox"/>	100UR01-PF03_X1	M20.0	BOOL	Light Start
<input checked="" type="checkbox"/>	100UR01-PF04_X1	M20.1	BOOL	Light Reset
<input checked="" type="checkbox"/>	110UR01-PF01_X1	M20.2	BOOL	Light Cycle start color 1
<input checked="" type="checkbox"/>	110UR01-PF02_X1	M20.3	BOOL	Light Cycle start color 2
<input checked="" type="checkbox"/>	120BT01-PF01_X1	M20.4	BOOL	Light button request door open/close mess position
<input checked="" type="checkbox"/>	130CL01-PF01_X1	M20.5	BOOL	Light button request open door NOK parts
<input checked="" type="checkbox"/>	140UR01-PF01_X1	M20.6	BOOL	Light output position - possibility
<input checked="" type="checkbox"/>	120UR01-QQ01_E1	M20.7	BOOL	Door mess position lock
<input checked="" type="checkbox"/>	RemoteControlON	M21.0	BOOL	Remote control ON
<input checked="" type="checkbox"/>	100UH01-TF11_A1	M22.0	BOOL	DUT1 in position
<input checked="" type="checkbox"/>	100UH01-TF12_A1	M22.1	BOOL	DUT2 in position
<input checked="" type="checkbox"/>	100UH01-TF13_A1	M22.2	BOOL	Test start

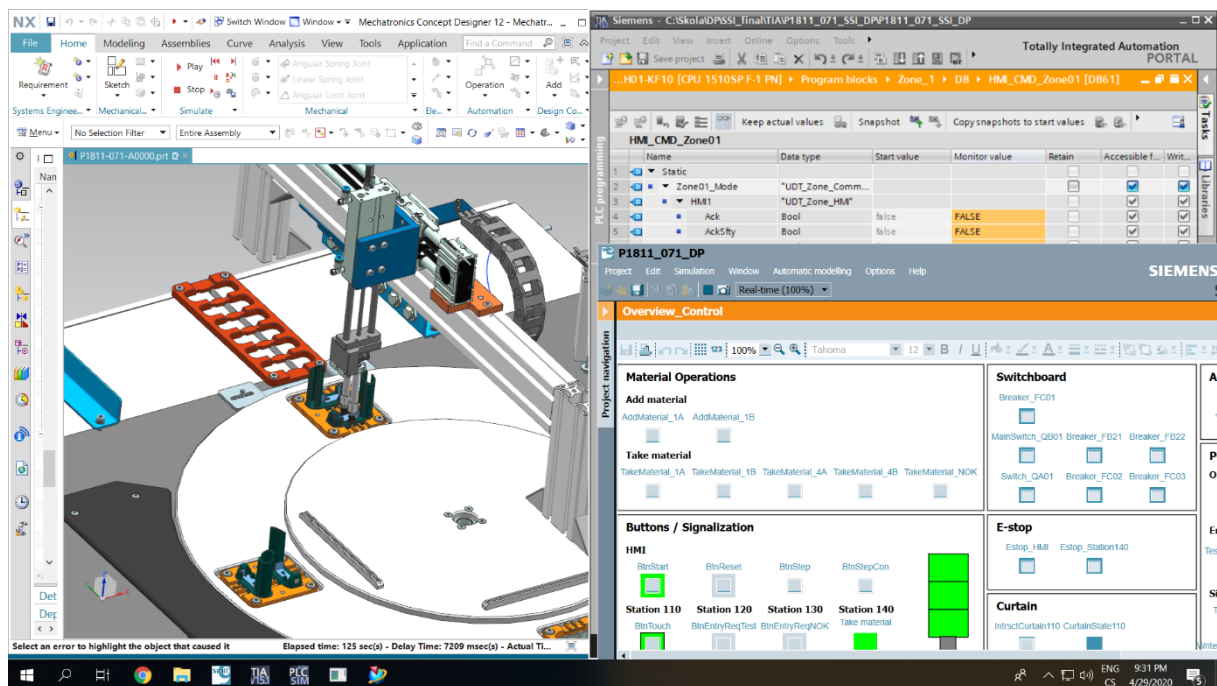
Obrázek E.2: Výměna signálů mezi nástroji PLCSIM Advanced a SIMIT zobrazena v SIMIT



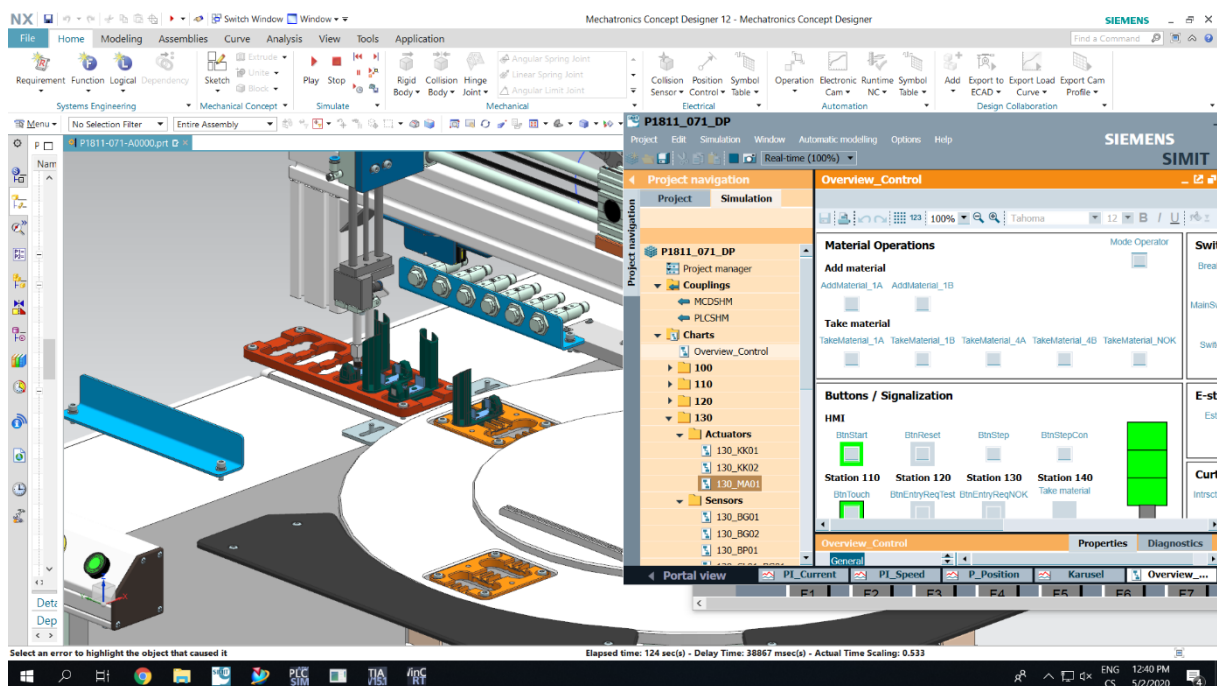
Obrázek E.3: Vytvořená instance PLC v PLCSIM Advanced



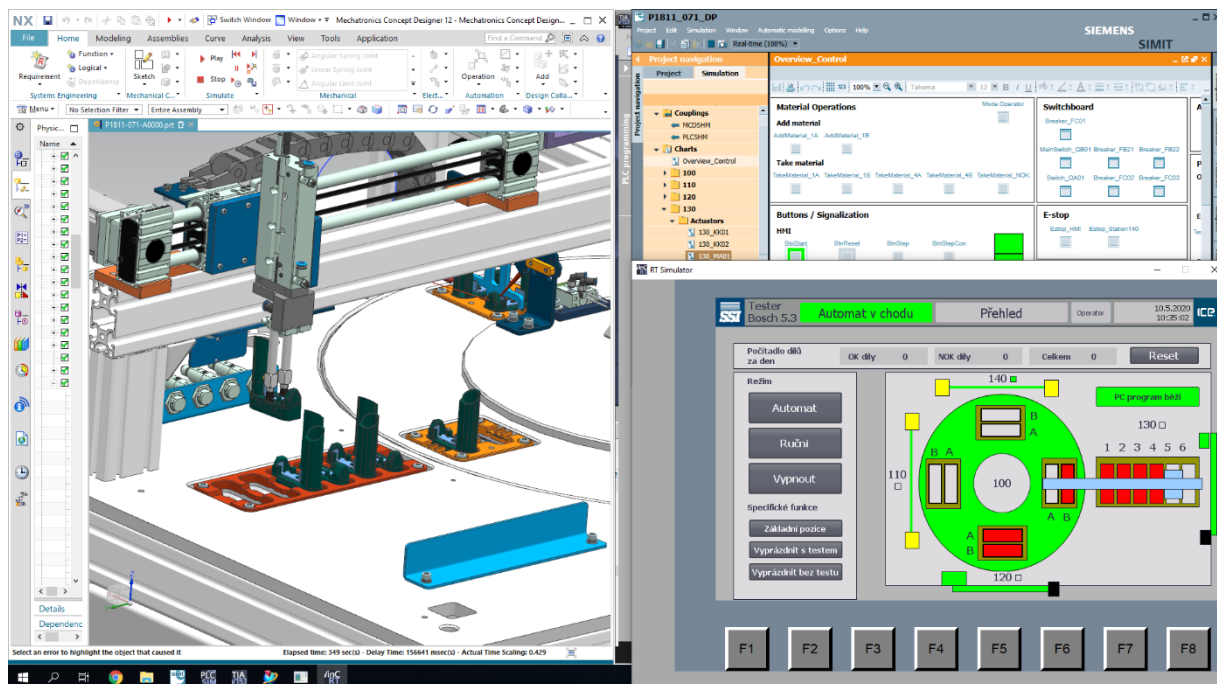
Obrázek E.4: Signálová výměna mezi nástroji při řízení motoru karuselu v SIMIT



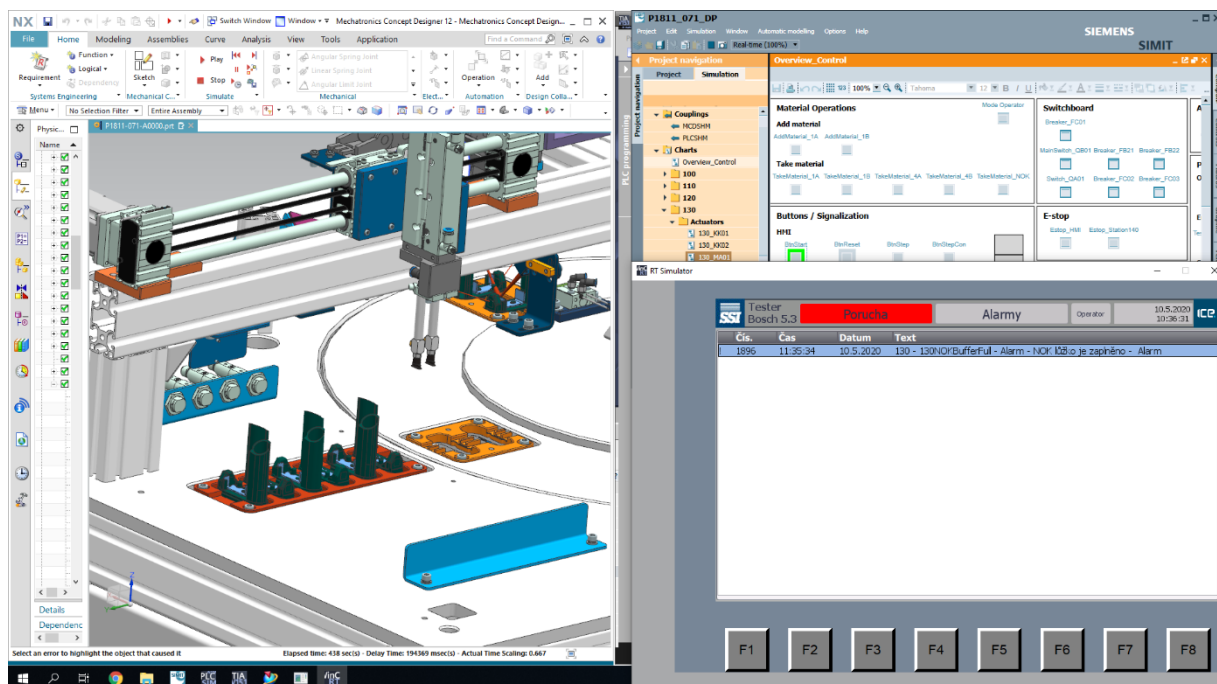
Obrázek E.5: *Testování řídicí aplikace na virtuálním stroji*



Obrázek E.6: *Testování řídicí aplikace na virtuálním stroji*



Obrázek E.7: *Testování řídicí aplikace na virtuálním stroji*



Obrázek E.8: *Testování řídicí aplikace na virtuálním stroji*



Task Manager										
File Options View										
Processes Performance App history Startup Users Details Services										
Name	Status	45% CPU	68% Memory	0% Disk	0% Network	100% GPU	GPU engine	Power usage	Power usage t...	
> Siemens.Automation.Portals (2)		0.1%	3,832.3 MB	0 MB/s	0 Mbps	0%		Very low		
> NX 12		18.4%	1,326.1 MB	0 MB/s	0 Mbps	91.8%	GPU 0 - 3D	Very high		
Siemens.Automation.Portals		0.1%	1,233.3 MB	0 MB/s	0 Mbps	0%		Very low		
Siemens.Simatic.Simulation.Ru...		11.3%	574.9 MB	0 MB/s	0 Mbps	0%		Very high		
Siemens.Automation.ObjectFra...		0%	569.3 MB	0 MB/s	0 Mbps	0%		Very low		
> SIMIT (32 bit) (3)		0.9%	119.6 MB	0 MB/s	0 Mbps	0.9%	GPU 0 - 3D	Low		
> Windows Explorer		0%	67.3 MB	0 MB/s	0 Mbps	0%		Very low		
> SIMATIC HMI Basic RunTime Si...		0.2%	59.4 MB	0 MB/s	0 Mbps	0.1%	GPU 0 - 3D	Very low		
> S7-PLCSIM Advanced Control P...		0.2%	38.3 MB	0.1 MB/s	0 Mbps	0.2%	GPU 0 - 3D	Very low		
> SQL Server Windows NT (32 bit)		0%	36.5 MB	0.1 MB/s	0 Mbps	0%		Very low		
> SQL Server Windows NT (32 bit)		0%	35.9 MB	0.1 MB/s	0 Mbps	0%		Very low		
> Node.js: Server-side JavaScript		0%	34.2 MB	0 MB/s	0 Mbps	0%		Very low		
> Microsoft Office Click-to-Run (...)		0%	33.7 MB	0 MB/s	0 Mbps	0%		Very low		
> NVIDIA Share		0%	32.0 MB	0 MB/s	0 Mbps	0%		Very low		
> Paint		0%	31.8 MB	0 MB/s	0 Mbps	0%		Very low		
> NVIDIA Container		0.4%	27.2 MB	0.1 MB/s	0 Mbps	0%		Very low		
> Node.js: Server-side JavaScript		0%	26.0 MB	0 MB/s	0 Mbps	0%		Very low		
> Desktop Window Manager		0.7%	26.0 MB	0 MB/s	0 Mbps	6.9%	GPU 0 - 3D	Very low		
Fewer details										End task

Obrázek E.9: Zatížení komponent PC během kosimulace

Příloha F: *CD-ROM*

- Elektronická verze diplomové práce ve formátu PDF
- Fyzikálně-kinematický model jednoúčelového stroje vytvořený v NX Mechatronics Concept Designer 12.0.1
- Elektrický a behaviorální model jednoúčelového stroje vytvořený v SIMIT V9.0
- Řídicí aplikace jednoúčelového stroje vytvořená v TIA Portal V15.1
- Aplikace pro výměnu dat mezi instancí PLCSIM Advanced a sdílenou pamětí vytvořenou ve Visual Studio 2019
- M-file s postupem návrhu regulátorů pro DC motor a model regulované soustavy vytvořený v prostředí v MATLAB Simulink
- Video s virtuálně zprovozněným strojem ve formátu MP4 (spuštění kosimulace a chod stroje v automatickém režimu, ovládání stroje v servisním režimu, reakce stroje na zaplnění NOK lůžka)